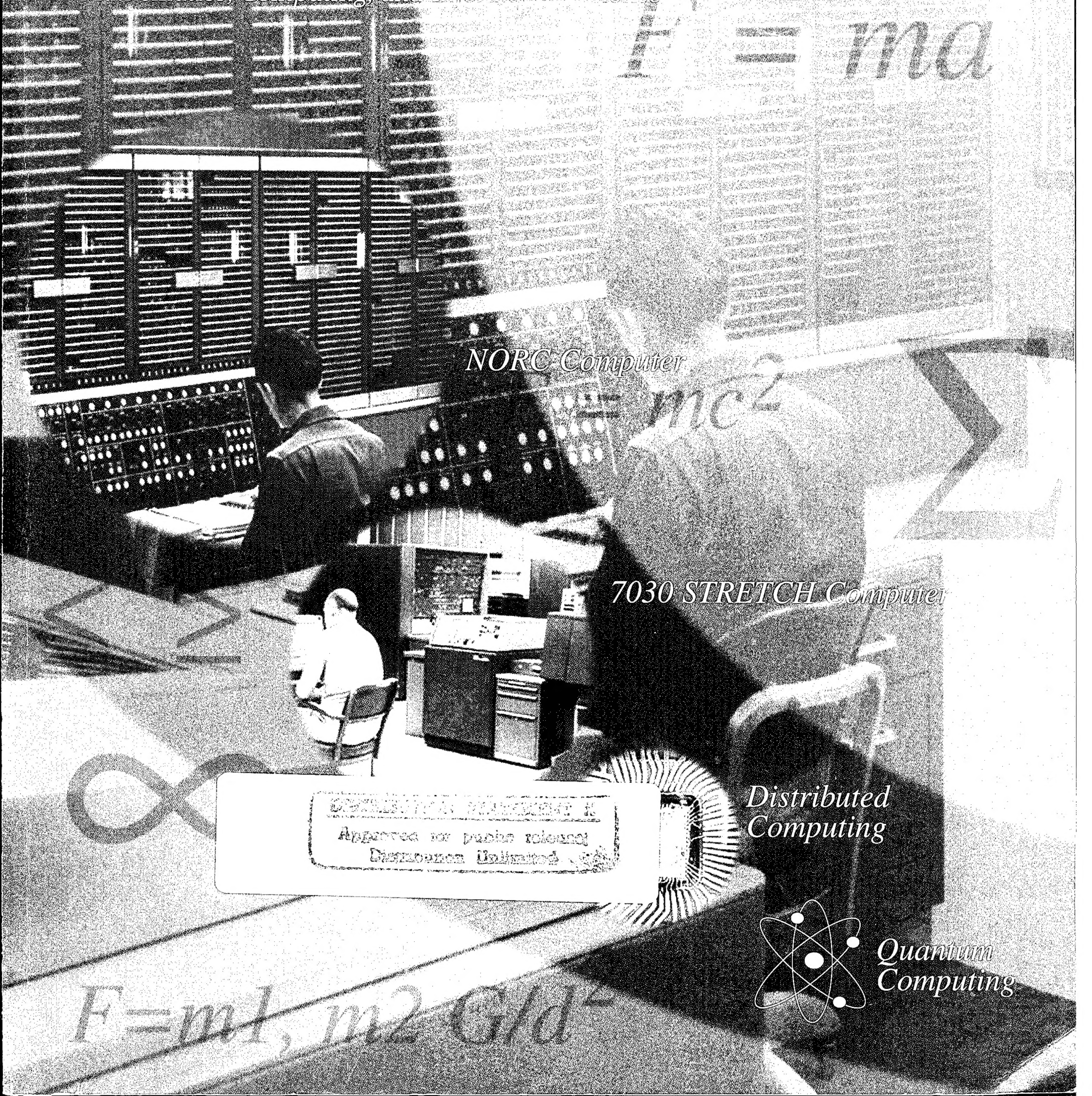
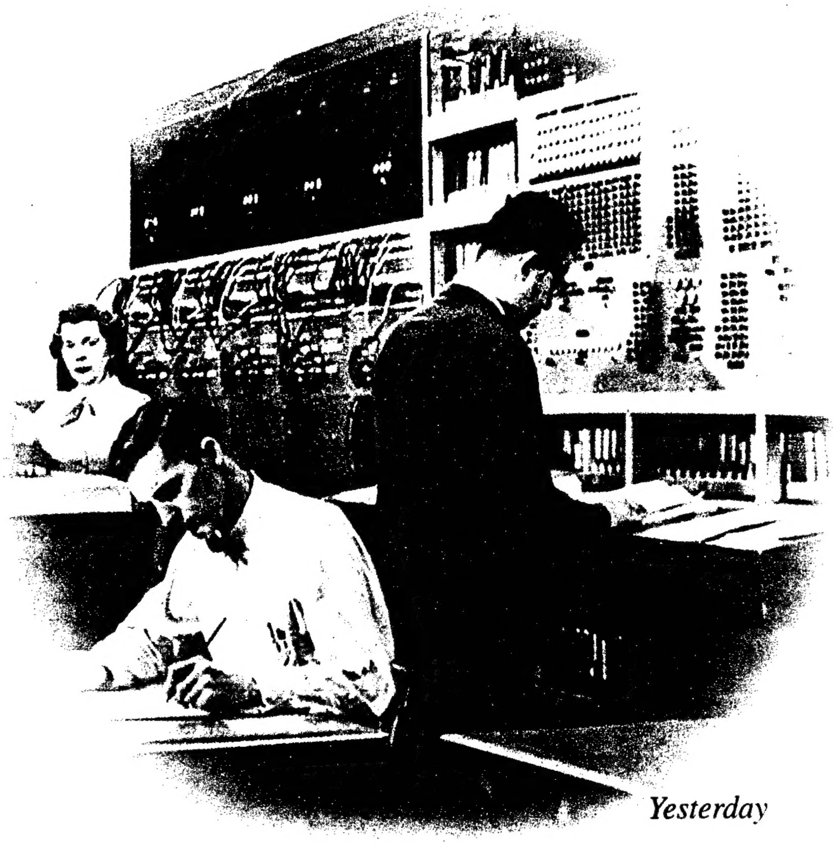
Naval Surface Warfa
Dahlgren Division

Technical Digest

Mathematics, Computing, and Information Science




Yesterday

The *Naval Surface Warfare Center Dahlgren Division Technical Digest* presents unclassified articles, contributed primarily by Division scientists and engineers, on selected research and development programs. The Dahlgren Division, under the leadership of the Naval Surface Warfare Center, provides research, development, test and evaluation, engineering, and Fleet support for surface warfare systems, surface ship combat systems, ordnance, mines, amphibious warfare systems, mine countermeasures, special warfare systems, and strategic systems. Please address any correspondence concerning the *NSWCDD Technical Digest* to: Dahlgren Division, Naval Surface Warfare Center, Technical Digest (Code E282), 17320 Dahlgren Road, Dahlgren, VA 22448-5100. Telephone: (540) 653-8921.

Today


About the cover: Figures in the shaded arc depict 40 years of computer evolution at NSWCDD. Descending from left to right across the cover are photographs of the Navy's Naval Ordnance Research Calculator (NORC) and the 7030 STRETCH Computer, and icons representing today's distributed computing (as evident in the AEGIS and TOMAHAWK systems) and tomorrow's quantum computing. Initially driven by ballistics and strategic needs, respectively, both NORC and STRETCH were disproportionately large considering their relatively limited functions. While computer hardware has since shrunk dramatically, its capability has increased exponentially, which permits the handling of ever more complex tasks and ever greater systems integration. This has revolutionized the Navy's systems and their use. Symbols superimposed on the background above and below the arc represent the mathematics and information science elements addressed in this issue. Our solid history and the ongoing innovations in these and related disciplines have enabled NSWCDD to stay at the forefront of mathematics, computing, and information science technology to better serve the Fleet.

NSWCDD/MP-95/95
Approved for public release;
distribution is unlimited.

Naval Surface Warfare Center
Dahlgren Division

Technical Digest

1995 Issue

System Applications

Guest Editors' Introduction

Harry E. Crisp, II, Richard A. Lorey, and W. L. McCoy

The Naval Surface Warfare Center, Dahlgren Division (NSWCDD) has significant technical responsibilities in major Navy programs that stem from strong capabilities in mathematics and computing technology. This issue of the Technical Digest traces the origins of these capabilities, presents ongoing research and technology, and provides some examples of applications to major programs.

Capability Evolution

The NSWCDD history is rooted in strong mathematical and computational capabilities, which positioned it to play key roles in Navy programs (see Figure 1). The origins of these capabilities are found in the 1918 mission of the Naval Proving Ground/Indian Head Lower Station to test Navy guns and ammunition, specifically the element of performing ballistics analyses. The mathematics capabilities that were established to accomplish this evolved and matured during World War II, driving a demand for computing machinery to automate the process. Ultimately, the first Navy computer was acquired and placed at the Dahlgren laboratory to support the need for advanced ballistics computation.

The continued evolution of mathematics and computing capabilities at NSWCDD has resulted in a number of advances in the state of practice in military computing technology. The contributions made by NSWCDD scientists and engineers mirror the evolution of computing technology—from the use of the largest, commercially available mainframes to support scientific and engineering computing at the Dahlgren laboratory; to the validation and application of military standard mainframes and minicomputers for embedded system requirements; to current applications of workstations, personal computers, and networks within systems; and to the support of Navy research, development, and engineering endeavors. The NSWCDD contributions to the field of military computing are documented in a number of technical reports and published papers.

Key contributions have also been made in the application of advanced mathematics to a broad range of Navy requirements. These include, for example:

- Exterior and interior ballistics
- Orbital mechanics
- Gravitational fields
- Earth tides
- Target estimation and prediction
- Signal processing
- Target identification
- Data fusion
- Fire control

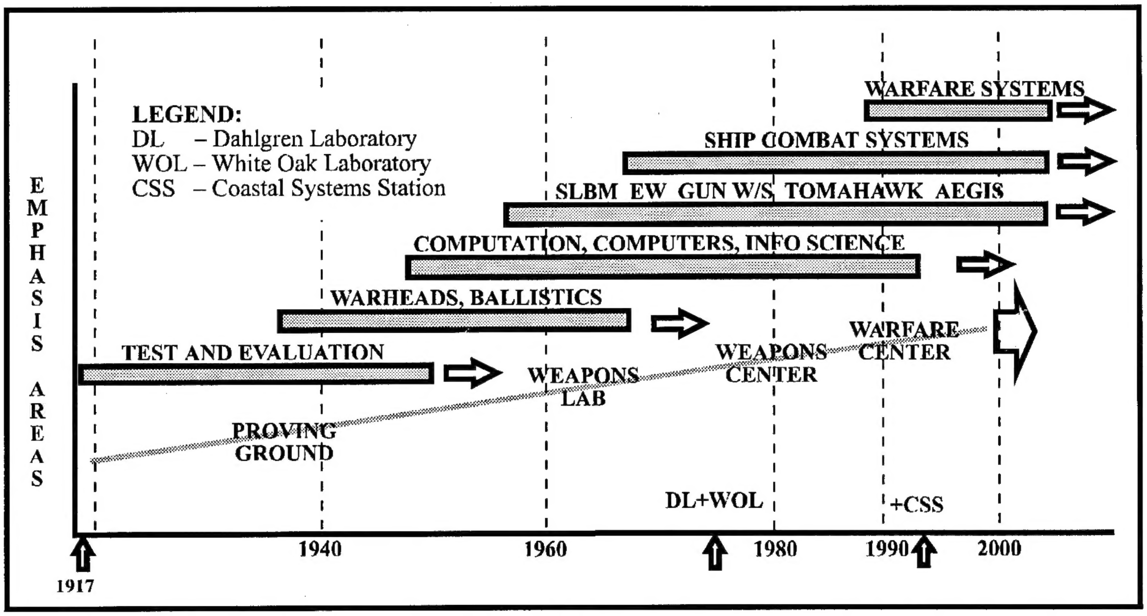


Figure 1. *History of evolution and revolution.*

- Weapons control
- Tactical decision aids
- The general area of scientific and engineering mathematics

These accomplishments are likewise documented in numerous technical reports and published papers.

NSWCDD Roles

The evolving Dahlgren laboratory foundations in mathematics and computing placed it uniquely in a position to play key roles in an ever broadening range of Navy and Department of Defense (DoD) programs. The ballistic computation capability provided the basis for involvement from the earliest stages of the Submarine Launched Ballistic Missile (SLBM) program through the current Trident II. This role, in turn, led to technical responsibilities in the application of computing technology to surface ship weapons systems. The experience gained from these assignments, coupled with an increasing base of technical expertise in shipboard weapons and sensors, led to broader roles in engineering weapons, sensor, intelligence, and combat systems for Navy ships. The current NSWCDD mission assigns broad responsibilities for engineering combat and weapons systems for the Navy. Along the way, NSWCDD has made significant contributions to other DoD, National Aeronautics and Space Administration (NASA), commercial, and Government agencies' programs.

NSWCDD currently is positioned to provide strong technical leadership for the design and development of next-generation Navy systems. The design of these systems will be based on the best available commercial computing technology and total ship systems engineering (TSSE) concepts. Ongoing NSWCDD research and development activities in both the application of mathematics and computing technology, and the underlying engineering processes are playing a significant role in the engineering decisions being made for a broad range of major program advancements, including the 21st Century Surface Combatant, the Advanced TOMAHAWK Weapon Control System (TWCS), the Ship Self-Defense System, and advanced baselines of SLBM fire control and the AEGIS shipbuilding program. NSWCDD is also investing in research in innovative computing concepts, including molecular computing devices and quantum computing, that could lead to profoundly different computer systems in the 21st century.

A Tour of the Digest

Previous issues of the *Technical Digest* have given some insights and evidence of the

significant capabilities of NSWCDD in mathematics and computing technology. This issue provides a more comprehensive view, beginning with the historical origins and tracing the evolution to modern capabilities. Examples of current themes in mathematics are presented, along with trends for the future in computing technology. Finally, examples of applications to current acquisition program requirements are provided.

Historical Background

Hughey leads off with an insightful review of the history of mathematics and computing technology at the Dahlgren laboratory. His article documents the origins of the ballistics analyses capabilities and the evolution towards advanced computational techniques and computing machinery. He also outlines the manner in which this resulted in key roles in Navy strategic and space systems and major shipboard systems. Green then discusses the evolution of digital computer technology in U.S. Navy surface ship combat systems, from the 1950s through the present time. He makes the observation that hardware, software, and design tools have evolved dramatically, yet the underlying problems to be solved and the basic approach have remained essentially unchanged. Pollard and Duren provide a view of the future in their article, which addresses a new framework for TSSE. They outline a family of backbone control structures that provide a means for mission teams to operate a ship as a coherent entity.

Mathematics

Parks discusses the theory of quantum computation as the basis for a more complete model for computing devices. He argues that the Universal Turing Machine, based on classical physics, is inadequate since the universe is quantum physical. He cites recent research that strongly suggests that quantum computing devices would provide computational power far exceeding that achievable by contemporary computing machines. Cawley, Hsu, and Salvino address a contemporary topic in nonlinear dynamics, or "chaos theory." They specifically present the Time Series Analysis developed by NSWCDD to support chaotic data analysis, which is observable in many physical processes. Crigler completes this section with his documentation of the creation and evolution of two computer libraries developed at NSWCDD. The first of these is the NSWCDD Library of Mathematics Subroutines (NSWCLIB), a specialized collection of general purpose mathematical software. It has achieved national and international acclaim in the scientific and engineering computing community. The second is the NSWCDD Library of Statistical Programs and Subroutines (STATLIB), a specialized collection of probability and statistics software.

Information and Systems Science

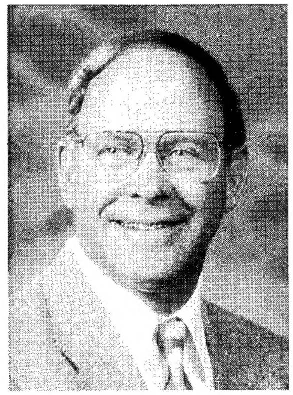
Crisp discusses the characteristics of large, complex systems and the need for an integrated design methodology to achieve performance, schedule, and cost objectives in acquiring future versions of these systems. The technical thrusts of the Office of Naval Research program in the Engineering of Complex Systems are described in the context of achieving a seamless flow of design activities across the systems development process and, also, "flowdown" into the application specialties. Masters presents the AEGIS shipbuilding program joint demonstration with the Advanced Research Projects Agency on the feasibility of inserting commercially available distributed computing technology in the AEGIS combat system. The results of two major experiments are presented. Tate, Boyd, Cullin, and Brizzolara provide a view of future generations of computing technology, based on molecular computing devices. They review NSWCDD's research efforts in the potential of biomaterials with desired properties for use in computational architectures, including characterizing and utilizing the optical properties of these materials. Batayte provides an insightful article on the growing NSWCDD capabilities in virtual reality research and technology and potential applications to decision support. The dual-use nature of much of the military computing technology is characterized in the article by Lorey, Solka, Rogers, Marchette, and Priebe.

In this case, research results in target identification by pattern recognition have been applied to mammographic computer-assisted diagnosis.

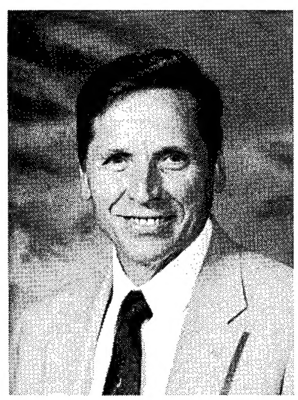
System Applications

Current themes in the application of advanced computing technology are presented in the articles contained in this section. Gates reviews the evolution of NSWCDD involvement in the SLBM program, including contributions to computational methods, computer languages, operating systems, and fire control system architecture. The knowledge and experience gained over the 40 years of involvement are now being used to propose innovative solutions for the SLBM fire control system of the future. The NSWCDD involvement in SLBM led to key technical responsibilities in the TWCS. Thomas, Horne, Sheehan, and Tripp review the history of the TWCS computing architecture development. The Advanced TWCS responds to new requirements for strike weapon control—a coordination based on a flexible (open) architecture. Moritz, in his article on mine countermeasures simulation, addresses the growing role of modeling and simulation in warfare systems analysis and design. Finally, Podlesny postulates the need for a "holistic engineering concept" to address needs such as rapid use of emerging technologies, team structured management, and use of rapid prototyping methodologies to deliver quality products.

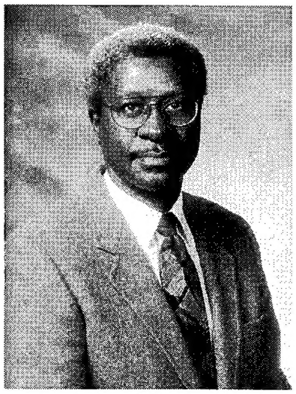
The Guest Editors



HARRY E. CRISP, II is Program Manager for Engineering of Complex Systems, an exploratory development program sponsored by the Office of Naval Research. He graduated from Clemson University in 1964 with a B.S. degree in electrical engineering. He received an M.S. and a Ph.D. in electrical engineering from Auburn University in 1969 and 1971, respectively. He has been employed at NSWCDD since September 1971. He has performed research in digital control systems, served in the MK 86 Program Office at the Naval Sea Systems Command, and has been the NSWCDD Program Manager for the MK 86 Research and Development Program. He has been Head of the Weapons Control Technology Branch, the Information and Control Technology Branch, and the NSWCDD Technology Base Program Office. He has also served as director of the NSWCDD Independent Exploratory Development Program. He is a member of the Institute of Electrical and Electronics Engineers, the American Society of Naval Engineers, and the International Council on Systems Engineering.



RICHARD A. LOREY earned a B.S. degree in physics in 1962 from the University of Pittsburgh and an M.S. and a Ph.D. in physics in 1967 and 1969, respectively, from the Georgia Institute of Technology. He has been at NSWCDD since 1969. At Dahlgren he has worked in weapons safety research, and range instrumentation development. He has also served as branch head of Electronic Warfare-Countermeasures, SLBM Targeting, SLBM Fire Control Development, and Space and Geodesy. He has received the Department of the Navy Meritorious Civilian Service Award. Currently, he is head of the Advanced Computation Technology Group in the Systems Research and Technology Department. His research interests include computational statistics and pattern recognition applied to both military use and medical imagery.



W. L. MCCOY is project leader for the Evaluation and Assessment Technology Task of the Engineering of Complex Systems Technology Program. He holds a B.S. degree in mathematics from Morehouse College and an M.S. degree in computer science from Purdue University. Most of his professional career has been spent in the Software Development Division of the Submarine Launched Ballistic Missile (SLBM) Program. He has performed and managed R&D in software and systems engineering technology and applied the results to SLBM, Federal Aviation Administration, and other Navy programs in the areas of software development and computer system performance engineering. He has also served on several DoD committees and study groups, performed a technology transfer assignment at the Federal Aviation Administration, and completed the Department of Commerce Science and Technology Fellowship Program. He is currently a member of the IEEE Computer Society, the Association of Computing Machinery, the International Council on Systems Engineering, and the New York Academy of Sciences.

History of Mathematics and Computing Technology at the Dahlgren Laboratory

Raymond H. Hughey, Jr.

Mathematics and computing technology at the Dahlgren laboratory evolved as a direct requirement of developing laboratory and Navy mission needs and responsibilities. Early mathematics capabilities were essential to performing ballistics analyses critical to the initial mission of the Dahlgren Proving Ground. The laboratory responded to the need for range tables and bombing tables in World War II by substantially extending exterior ballistics theory and computational techniques, and by pursuing the acquisition of new technology capabilities in computing machinery. The foresight and technical expertise of some of Dahlgren's key scientists enabled the extension of knowledge to meet Navy needs in developing new strategic and space systems. The challenges of these systems demanded new foundations in technical expertise and systems experience, which supported Dahlgren's research and development (R&D) in other major Navy shipboard systems, such as AEGIS and TOMAHAWK. Scientists and mathematicians at the laboratory continue performing research to extend the state of the art and to develop fleet-ready implementations using advanced technology.

Background

The Naval Proving Ground/Indian Head Lower Station at Dahlgren was established in 1918 to remedy increasingly restrictive range limitations and hazards for gun testing at Indian Head. The Dahlgren location was selected for its uncongested access to a relatively wide and straight portion of the Potomac River, which could provide a 30,000-yard range for testing heavy guns, and for its low cost. Congress approved a bill that permitted the President to acquire the land for the new Proving Ground and the first tract of land was obtained by Presidential Proclamation in June 1918. The initial use of the Proving Ground came quickly, with a successful firing of a 7-inch, 45-caliber, tractor-mounted gun on October 16, 1918 (see Figure 1). In late 1918 the decision was made to name the station in honor of Rear Admiral John Adolphus Dahlgren for his prominence in ordnance development and his role as the father of modern ordnance. Additional land for the Dahlgren station was provided by other Proclamations of the President in November 1918 and March 1919.[1]

One part of the initial primary mission of the station included the testing of Navy guns and ammunition (inert only, at the beginning) to obtain trajectory data of projectiles using range-table firings of guns and to obtain other ballistic data. Mathematics and computing technology at Dahlgren has its roots in this earliest mission. The ballistic coefficient was the primary parameter, and most of the work from this time through the next three decades was related to the Gavre retardation function.

Dr. L. T. E. Thompson came to Dahlgren in 1923, in the newly created position of Chief Physicist, to perform analysis related to experimental testing. He apparently performed his work entirely by hand and without any staff until 1935 when several more professionals arrived to assist him. His work in

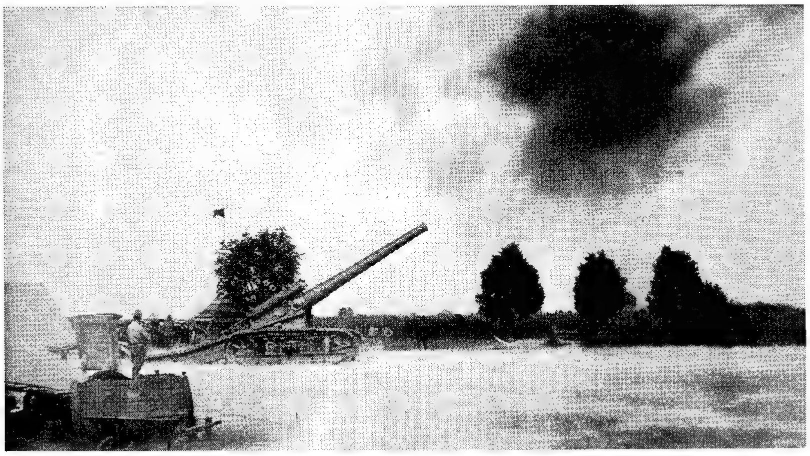
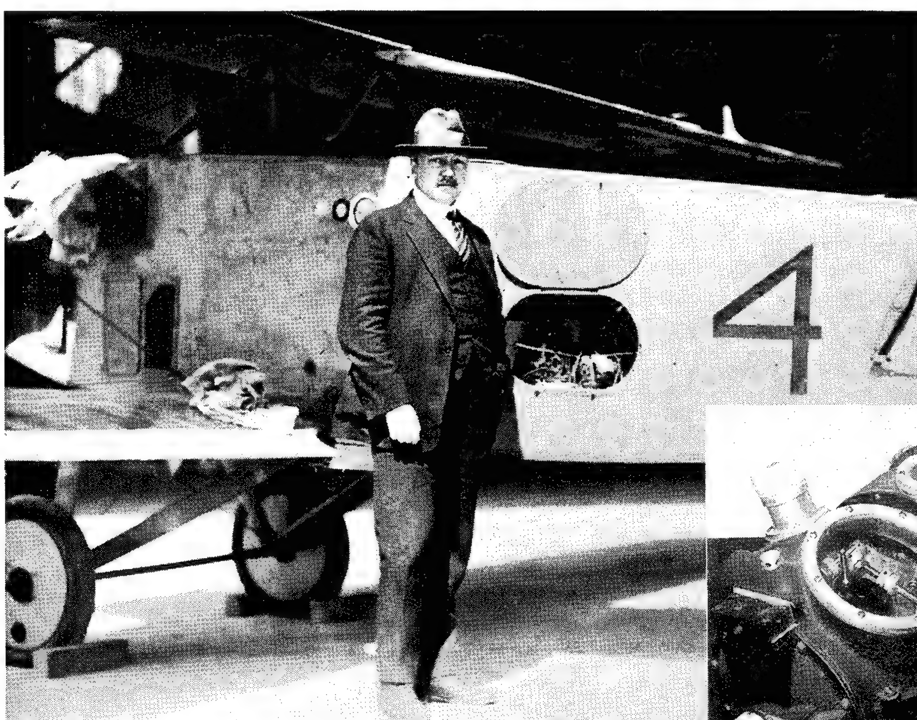


Figure 1. *Dahlgren Proving Ground was used for the first successful firing of the 7-inch, 45-caliber, tractor-mounted gun in October 1918.*

ballistics studies and investigations conducted at reduced scale revealed that penetration results were scalable. The economies resulting from his analyses permitted the continued performance of R&D at the Proving Ground during times of severely reduced military spending. He and his staff performed all analyses and computations relating to exterior ballistics, velocity measurements, hardware development, and test results.[2] Full-scale flight testing of the Norden bombsight was performed at Dahlgren from the 1920s through World War II (see Figure 2), and Dr. Thompson performed critical analysis for the bombsight in the 1920s and 1930s. In referring to his work in this area, Rear Admiral Boynton L. Braun stated:

He was really the brain down here for the mathematics work . . . we never had a genius in mathematics and physics that gave us the knowledge that he had, so we depended



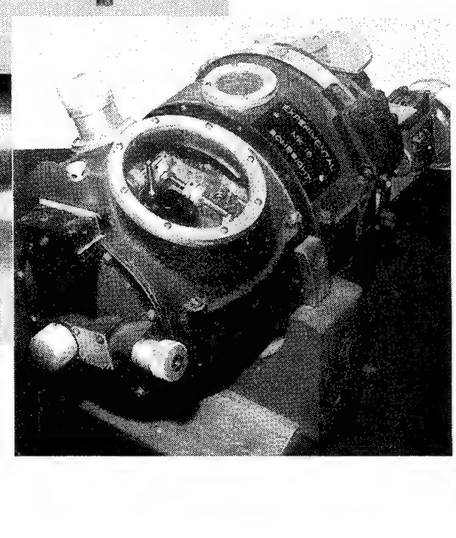


Figure 2. *Carl Norden with one of his bombsights shown installed in aircraft. Full-scale testing was performed at Dahlgren from the 1920s through World War II.*

on him for a lot of the analysis. Of course they didn't have computers in those days. The arithmetic computation had to be done in longhand.[2]

Dr. Thompson's foresight and perception of the need for scientific capability and work within the Navy establishment were rare in his time, and laid the foundation for Dahlgren's progression from pure proof and test work to the full scope of a research, development, test, and evaluation (RDT&E) laboratory.[2]

The First Critical Change

Although Dr. Thompson and his team had performed many gun, bomb, and unguided rocket trajectory and ballistic analyses over the years, the responsibility for computing ballistics tables remained with the Bureau of Ordnance in Washington until after the start of World War II. The war created an urgent need for greater capacity in exterior ballistics, and the view held that it was hard to do useful mathematical work in Washington. In June 1942, Dahlgren was tasked with the responsibility for range-table and bombing-table computations for all of the Navy's guns, unguided rockets, and bombs. This created the first organized upsurge in mathematics and computing at Dahlgren.

The computational requirements were staggering. Trajectories needed to be computed for all usable conditions for each gun-projectile combination to determine the range tables. Two simultaneous differential equations representing the second derivatives of range (horizontal position) and altitude were numerically integrated from time of firing to impact. The initial velocity was obtained from test firings, and gun jump had to be accounted for; gravity was commonly represented by a standard constant value. A standard definition of the atmosphere was usually utilized. The ballistic coefficient ($\beta = \frac{W}{C_D A}$, W = weight, C_D = drag coefficient, A = base area), based on range observations, was used to provide the drag as a function of velocity. Only two or three degrees of freedom were used in the integration. So the drift from aerodynamic forces on the projectile as it was affected by gravity and the gyroscopic response of the rotating projectile did not come out of the integration but were added to the range tables based on empirical equations based, in turn, on firing data. The antiaircraft range tables required computed data all along the trajectory, not just at the end points, and they increased trajectory integration volume requirements by more than an order of magnitude.[2] Dr. Charles C. Bramble, Senior Professor of Mathematics and Mechanics at the U.S. Naval Academy Postgraduate School (then at Annapolis), was assigned to Dahlgren to lead this effort—this was the beginning of the Computation Laboratory. At the time Dr. Bramble arrived at Dahlgren, he stated that he found:

. . . only two desk-type calculators in the place and two mathematicians to operate them.[1]

He immediately put in a request for five additional machines. Some of the scientists and engineers who were to have a major impact on the work at Dahlgren over the years joined the lab during this period. For example Dr. Charles J. Cohen and Mr. David R. Brown, Jr., came aboard in the 1943-1944 time period, along with other Naval Officers educated in science and engineering. In these early days, the numerical integration of trajectories was performed by desktop mechanical calculators. In 1944 and 1945, some 58 Navy Waves were stationed at Dahlgren specifically to perform manual trajectory computations and ballistics measurements. The desktop calculator effort was supplemented by a contractual effort on Bush differential analyzers at the Massachusetts Institute of Technology (MIT).

The MARK II and MARK III

Digital computing machines were beginning to be developed, and Dahlgren saw them as the solution to its monumental computing problem. Dr. H. H. Aiken of Harvard, in cooperation with IBM, began design and construction of the MARK I Automatic Sequence-Controlled Calculator in 1939, completing it in August 1944. This has been hailed as the first general purpose digital computing machine ever completed. Scientists at Dahlgren were following this work and recognized that such a machine could relieve the overwhelming computing burden that was building.

As a result of Dahlgren's recommendations, in March 1945 the Bureau of Ordnance entered

into a contract with Harvard to develop the MARK II computer, also known as the Aiken Relay Calculator.

The MARK II (shown in Figure 3) cost $840,000, occupied 4000 square feet of floor space, and contained 13,000 relays. In addition to his team of engineers who designed the machine, Dr. Aiken added mathematicians to develop programming and coding capabilities for the machine. (Ralph A. Niemann was one of these and came to Dahlgren with the machine in 1947. Later, he headed up the computation and mathematics effort for almost 25 years before retiring in 1979.) Dr. Aiken also hired ten technicians from the Boston area to assist with construction and operation of the MARK II, with the understanding that they would go to Dahlgren to work with the machine after it was completed. Some of these technicians worked at Dahlgren for more than 30 years.

A short while before the machine was declared completed, disassembled, and shipped to Dahlgren, one of these technicians, Bill Burke, was searching for the cause of a computation error in the machine on the afternoon of September 9, 1947. He finally traced the error to a moth caught in one of the relays. Mr. Burke removed the moth, checked to determine that the computer then worked properly, and taped the moth into the daily computer maintenance log with an annotation of the repair (see Figure 4). From that time on, the engineers and technicians referred to finding and removing a fault in the computer or a program as "debugging." This "bug" is widely credited as being the original computer "bug." RADM (then Lieutenant) Grace M. Hopper, who became associated with Harvard and Dr. Aiken as a Naval Reserve officer in 1944, and then joined the Harvard faculty as a research fellow in 1946, worked with the MARK II design and development team. She told the story of the bug (in varying forms) to hundreds of audiences over the years, and greatly added to its fame. The log with the bug was kept at Dahlgren for many years until, in 1993, it was presented to the Smithsonian Institution, at its

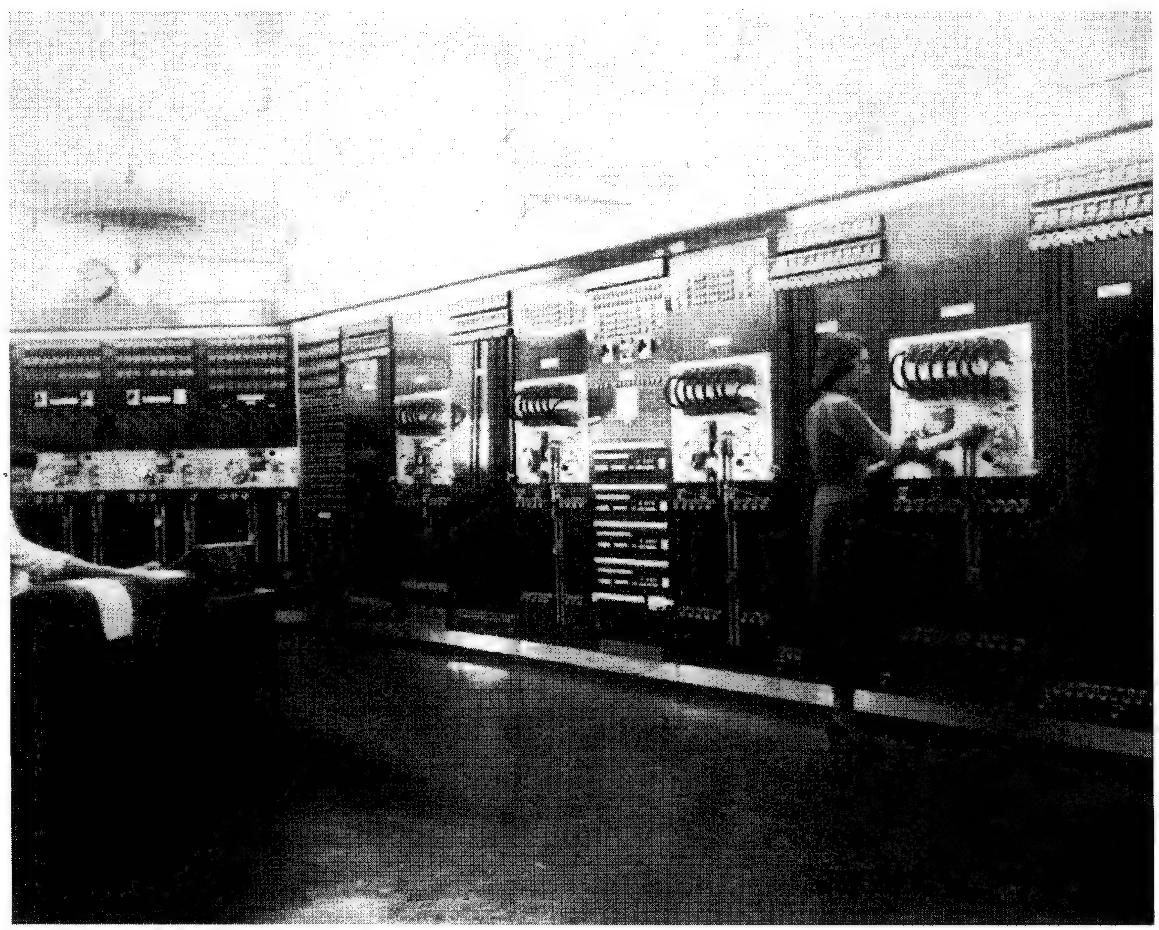


Figure 3. *The MARK II Aiken Relay Calculator.*

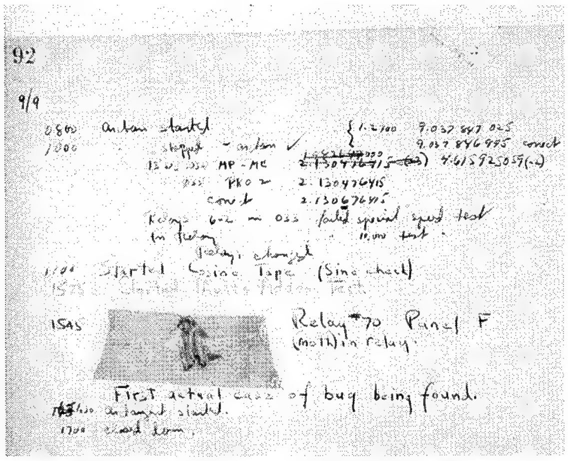


Figure 4. *The famous computer "bug."*

request, to be stabilized against further deterioration and made available for special exhibits. In 1994, the principal TV network in Japan sent a team of photographers to Washington just to videotape the bug for inclusion in their TV special on the history of computers.

Shortly after the contract for the MARK II had been awarded, Dahlgren recommended that another computer—this one an electronic calculator—be constructed even before completion of the MARK II. In April 1946, a contract was awarded to Harvard for the MARK III Magnetic Drum Calculator (see Figure 5), and Dr. Aiken and his team began work on the new machine while completing work on the MARK II. Before the MARK III left Harvard, it had already become apparent that there were adjustment problems with the magnetic drum and that the read-write heads were going to be a maintenance problem. The machine, which was delivered to Dahlgren in March 1951, was expected to be 10 times faster than the MARK II, but the reliability of the drum access was so poor that programmers were forced to code in mathematical checks and duplicate code in order to have any assurance that results were correct. This was quite a disappointment to Dahlgren scientists and engineers who had experience with the high reliability of the MARK II. With extreme coding checks and redundancy, together with refined maintenance procedures and the

Figure 5. *MARK III Magnetic Drum Calculator.*

incorporation by Dahlgren engineers of some minor design changes, some useful output was obtained from the MARK III after December 1952. However, the effective speed was far below the expected speed. According to Ralph Niemann:

> *. . . it was the unanimous verdict of the Dahlgren staff that the MARK III had not lived up to its original expectations. However, it was one of the first magnetic drum computers and the experience with its design, construction, and operation contributed to the knowledge of computer technology.*[3]

Post World War II Ballistics

The great increase in requirements for ballistics analysis and computations that began in World War II did not end with the war. In fact, the requirements continued their rapid growth. New bomb developments added to the demand. For example, the Low Drag Bomb demonstrated aerodynamic instabilities mostly related to the dependence of forces and moments on roll orientation. Considerable analyses of the instabilities were performed, and especially those related to the roll lock-in phenomenon. Of special significance to later work at Dahlgren were the advances in unguided rocket work after World War II. Ballistic theories to support aerodynamic performance and aiming data tables for them were inadequate. Most computations for the motion of projectiles had employed only the drag component of the aerodynamic forces. This is totally unsuitable for predicting the stability characteristics of rockets. Analyses at Dahlgren in the decade following the war, relating to the effects of thrust misalignment, wind deflection of the velocity vector during burning, rocket jet effects on aerodynamic characteristics, and the rocket's transverse angular velocity (primarily from gravity), which develops after it leaves the launcher, resulted in major advancements in ballistics theory and practice.[2] These advances required the collection of quality data from test firings, wind tunnels, and spark ranges, together with theoretical analyses and trajectory computations. The fascinating story of a prime example of this, relating to the previously unexpected instability of the 12.75-inch-diameter antisubmarine rocket (called Weapon A) as a result of the Magnus moment, is eloquently told in Reference 2.

The First Six-Degree-of-Freedom Trajectory Simulation

Dr. Cohen led a Dahlgren effort to develop a trajectory simulation, based on a set of differential equations formulated by the Naval Ordnance Test Station to represent the simultaneous translational and angular motions of the 12.75-inch rocket. This was among the earliest research projects for the MARK II computer. The numerical integration process used for computing the six-degree-of-freedom trajectory was the method of Runge-Kutta. This simulation, which was completed in 1950, is believed to be the world's first operational six-degree-of-freedom trajectory simulation.[2]

The Naval Ordnance Research Calculator (NORC)

Before the MARK II and MARK III were completed, the Navy was already exploring the next step forward in computers. Personnel at the Naval Ordnance Laboratory (NOL) in White Oak, Maryland, were major players in the initial investigations, and they contacted IBM about what capability was technically feasible at the time. After NOL discussions with the Bureau of Ordnance, the Bureau appointed a committee to consider the Navy's need for an advanced computer. Dr. Bramble from Dahlgren and Dr. R. J. Seeger from White Oak were members of that committee. The committee met with IBM and Remington Rand, and consulted with many experts, including Dr. John Von Neumann, who was building the Institute for Advanced Study (IAS) machine at Princeton, New Jersey. The committee then recommended that a contract be let with IBM to build the NORC. Another committee was appointed, with members from Dahlgren, White Oak, and the Bureau, to track IBM in its design and construction of the machine. The decision to place the machine at Dahlgren or White Oak was deliberately delayed until about eight months before the machine was ready for delivery in order to retain full interest by both

sites and, thus, assure that the Navy would obtain the best computer possible.

The White Oak representative on this committee was Dr. Harry Palachek, and the Dahlgren representative was initially Mr. Donald Heiser, who was replaced by Mr. Ralph A. Niemann in 1952. Both the Dahlgren and White Oak members were concerned about the coding system and the difficulties placed on the programmer. As a result, many features were incorporated into the design to ease the coding burden. Previous machines had been designed for hardware efficiency without considering coding efficiency. As a result of the experience with the MARK III, the Dahlgren representatives strongly believed in the need for self-checking features in the machine. Mr. Heiser first expressed this view to the committee; then Mr. Niemann and Mr. James R. Gros presented what Dahlgren thought was necessary in order to have confidence that the computer results had no machine errors. The Bureau permitted Dahlgren to make the case to IBM, who initially objected on the grounds that such a capability would require duplication of the hardware, and that increased hardware would reduce machine reliability. Dahlgren proposed techniques, such as redundant bit-count checks on transfer of numbers, that could detect errors without a lot of additional hardware. IBM agreed they would attempt to design self-checking into the machine without much hardware duplication and came back with a proposal that added about \$300,000 to \$400,000 to the original cost estimate of \$1,300,000. The Bureau of Ordnance agreed to this and directed IBM to proceed designing the NORC with error checking built in.

In retrospect, this has been viewed as money well spent as it assured the accuracy of results with very low probability of an undetected machine error. The later use of this machine for computing presetting data for the POLARIS missile made this an especially important step.[3] The Bureau of Ordnance announced its decision about the location of the machine about eight months before completion, and preparations were made at Dahlgren to receive it. The NORC (see Figure 6) was the most powerful machine in the world when it was completed,

Figure 6. *The Naval Ordnance Research Calculator (NORC).*

and several years elapsed before other machines exceeded its capability.

Computers After the NORC

Increasing computing demands for POLARIS, the buildup of work in determining the earth's gravity field, and processing associated with space surveillance soon overloaded the NORC; first one, and then a second IBM 7090 were obtained to supplement the capability. The capacity of these machines was almost immediately saturated, and Dahlgren was permitted to enter into negotiations with IBM for a STRETCH (IBM 7030) computer to be delivered in 1962. The arrival of this machine represented a large step forward in computing power (though less than had been anticipated), but the demand for computational capability was growing even faster. Increased demands for POLARIS, then the order of magnitude increase required by POSEIDON (see Figure 7), the first ballistic missile equipped with Multiple Independently-Targetable Reentry Vehicles (MIRVs), and the growing demands by an expanding astronautics and geodesy program soon absorbed all the capability the STRETCH could provide. Preparations to procure the next step in computing capability began in 1965. However, the government regulations and procedures in that time period were in such a state that, except for the addition of an IBM 360/40 for management support in 1965, the next major computer, the CDC 6700, wasn't installed until 1972—long after the previous capability had been saturated. This machine has been succeeded by a series of increasingly capable machines to provide for the growing need. The mainframes that have been used at Dahlgren from the CDC 6700 to the current Cray machines are listed in Table 1.

The Ballistic Missile Impact

There has been a popular belief that Dahlgren was given its role in POLARIS because it had the

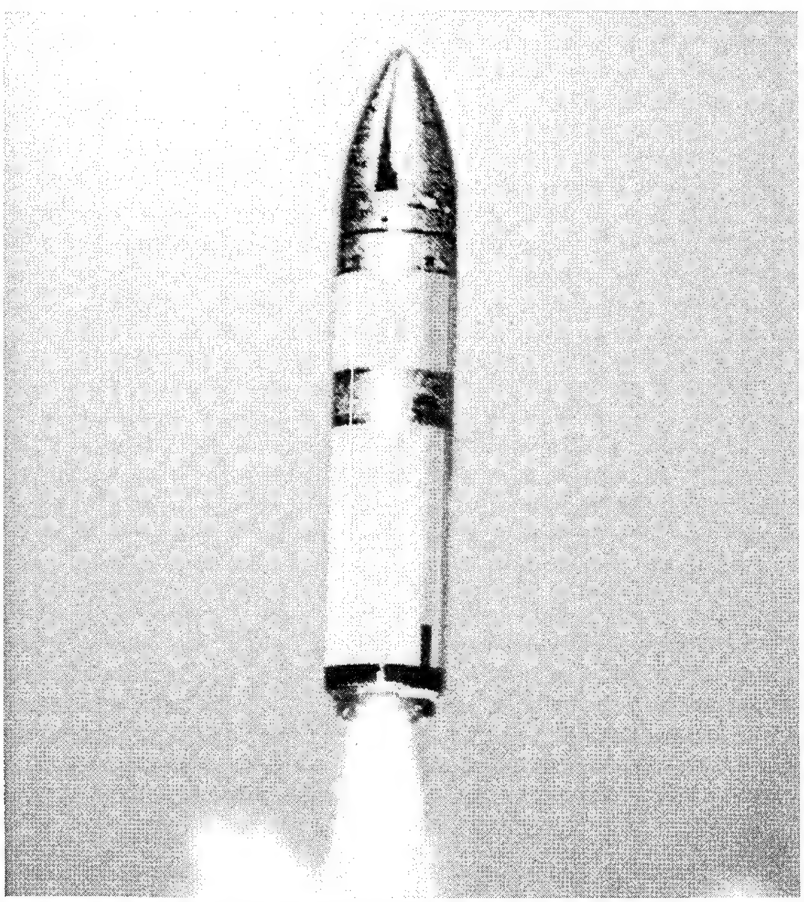


Figure 7. *POLARIS and POSEIDON Ballistic Missiles increased the demands for computing and characterizing earth's gravity fields.*

Table 1. *Control Data Corporation and Cray Research, Inc., Mainframes at Dahlgren*

Mainframe *	Period	CPUs	Speed (MFLOPS) **	Memory (Mbytes)***	OS
CDC 6700	72-85	2	0.48	1.3	SCOPE
CDC CYBER 74	76-81	1	0.32	1.3	SCOPE
CDC CYBER 760	81-85	1	2.6	2.6	SCOPE
CDC CYBER 180-825	83-93	1	0.19	20	NOS & NOS/VE
CDC CYBER 170-865	84-87	1	3.1	20	NOS
CDC CYBER 170-865	84-87	1	3.1	20	NOS
CDC CYBER 170-875	84-93	1	4.8	30	NOS
CDC CYBER 170-875	87-92	1	4.8	30	NOS
CDC CYBER 170-875	87-93	1	4.8	40	NOS
CDC CYBER 180-995E	89-pres	2	12	192	NOS & NOS/VE
Cray Y-MP2E/116	92-pres	1	161	128	UNICOS
Cray Y-MP2E/116	92-pres	1	161	128	UNICOS
Cray EL98/2-256(S)	95-pres	2	32	256	UNICOS

* - Both CDC CYBER 170-865s were field upgraded to CDC CYBER 170-875s in 1987.

** - Speed is in MFLOPS for Linpack with n=100 based on the paper "Performance of Various Computers Using Standard Linear Equation Software" by Jack J. Dongarra of the University of Tennessee dated 22 Jul 1993.[4]

*** - Memory is measured in 6-bit bytes prior to the CDC CYBER 180-995E, and in 8-bit bytes from the CDC CYBER 180-995E onward.

Navy's largest and fastest computers and that it had responsibility for Navy range tables. This is not the case. At the beginning of the effort to create a sea-based ballistic missile system, the Navy labs were given the opportunity to present their capabilities to the director of the Special Projects Office to see what they could contribute to POLARIS. Dahlgren's capabilities were ably presented by Dr. Russell Lyddane and Mr. Ralph Niemann, including the principal role in ballistics analysis, unparalleled computer capability, and leadership in six-degree-of-freedom missile flight simulation. However, no tasking to Dahlgren came as a result of this.

Building on the expertise that had been assembled over the years in exterior ballistics, Dr. Charles J. Cohen and David R. Brown were primarily responsible for forging Dahlgren's role in the Fleet Ballistic Missile (FBM) work. Their influence was the result of Dr. Cohen's foresight in recognizing the Navy's need before anyone else (as he did in many areas throughout his career) and his and David Brown's efforts in showing the Special Projects Office their ability to anticipate and fill technical requirements for the system. The first task they managed to get was for a reentry study for the Missile Branch. Then Dr. Cohen made the first guidance presetting studies for real operational conditions using the Q matrix that had been developed by MIT, and the results were presented at a technical coordination meeting at Lockheed. Mr. David Gold, the

chief engineer for the Guidance and Fire Control Branch at Special Projects, recognized the demonstrated capability to solve real system problems and decided to use Dahlgren in an advisory capacity. This gave Mr. Brown and Dr. Cohen the channel to determine and analyze problems and technical issues and to present results. They made the most of this opportunity and, during 1957, established Dahlgren as a vital player in the technical effort. Each piece of responsibility that Dahlgren gained over the years was vigorously contested by other organizations who also wanted the job, but Dahlgren prevailed by demonstrating the best technical capability and the ability to deliver the products.

Under the leadership of Dr. Cohen and Mr. Brown, a ballistic missile-inspired field of applied science called geoballistics (the word coined by Mr. Brown) emerged. Earlier, Dr. Cohen had developed an expression for the earth's gravity field in rectangular coordinates, having recognized that it would be needed for long-range ballistic missile trajectory computations. He also realized that expressions for the shape of the earth would be needed, that missile system accuracy would be limited by the representation of the earth's gravity field, and that missile guidance system initial erection and alignment would be affected by deflections of the vertical. Work was begun to address all of these problems. The scope of system and environment understanding, and the physical and mathematical representations required to determine the numerical quantities to be preset into the guidance system prior to launch, presented a monumental technical challenge. Among the most important achievements of this early team were its recognition, definition, and advancement of the technologies needed to put warheads on target.[2] The developing POLARIS team at Dahlgren pioneered technical solutions and computer implementations to meet these requirements and won the responsibility, held to this day, to be the technical source of all presetting data for the Submarine Launched Ballistic Missile (SLBM) program. Aiming data for the initial operational POLARIS was to be provided in the form of target cards, each usable from a 20-mile square of the ocean to a selected target area. It was determined that it would take over 40 years of NORC time to compute, using trajectory methods, all the aiming data for the card deck required for the first submarine to go on patrol. Therefore, worldwide grids of aiming data were computed and functionalized in terms of launch variables, and the functions, in turn, were evaluated for the conditions needed for each card.

With the POLARIS A2 Missile came the decision to equip the submarines with fire control computers to perform the aiming data computations. Dahlgren was assigned the responsibility to determine the problem, or mathematical formulation, to be solved in fire control, and for developing the fire control computer programs to implement it. This was the first fire control software development within the laboratory—and within the Navy. The increasingly demanding requirements of that responsibility (brought about by the extended missile ranges), the introduction of the MIRV capability, and the successively tightened accuracy requirements, compelled continued advancements in mathematical and computing techniques.[5]

POSEIDON, with the new concept of MIRVs and tighter accuracy requirements, brought enormous challenges. In the early feasibility-study stage of the program, Dahlgren developed the basic guidance method and presetting optimization approach that enabled fulfillment of system accuracy goals and resulted in a tractable fire control solution. The approach also provided patrol and targeting flexibility. Even with this contribution, the fire control problem was more than an order of magnitude more complex than that of POLARIS. Cramming this into the POSEIDON MK 88 Fire Control Computer, which was basically the same as that used for POLARIS with some upgrades recommended by the lab, was quite an achievement. A unique operating system called the POSEIDON SUPERVISOR was developed to help accomplish this feat. A considerable effort was put into developing some complex computer tools to prepare the targeters at Omaha and the Commanders in Chief (CINCs) to deal with the complexities of a MIRV system.

The next challenges came with the development of the TRIDENT I Missile, with its

increased range and tighter accuracy requirements. One feature introduced to help meet these requirements was a stellar sensor to permit inflight correction for accuracy improvement. Dahlgren developed star catalogues and star selection algorithms, as well as performed computations for optimized stellar-weighting matrices. It became apparent that the additional complexities were too much for the current fire control computer and that greater capability would have to be provided. The laboratory played a major role in determining the design characteristics and architecture for this new computer. The lab team also developed a new language, the Trident Higher Level Language (THLL) and an associated compiler. A real-time operating system and debug tools, which reflected the state of the art in computer science at that time, were also developed at Dahlgren. This was the first major Department of Defense (DoD) system for which the software was developed using the top-down structured approach. Its successful performance and very low maintenance requirements are testimony to the wisdom of that decision. These efforts were revolutionary in shipboard fire control.

The TRIDENT II system brought the challenge of a much tighter accuracy requirement. Dahlgren was able to develop new techniques to account for the high-frequency gravity effects, to develop an approach to obtain a more effective stellar-weighting matrix, and to improve the compensation of atmospheric effects on reentry vehicles. The system could not have achieved its accuracy goals without all of these contributions.[6]

Remarkable innovations have been made in the following areas to meet the needs of these evolving systems:

- Numerical analysis
- Trajectory computation
- Environmental-effects representation
- Fire control computer system architecture
- Operating systems
- Software development

Astronautics, Geodesy, and Space Systems

It was primarily from the efforts to support the ballistic-missile geoballistics requirements and from the incitement to pursue space sciences after the Soviet launch of Sputnik I in 1957 that the work in geodesy and astronautics flourished. By 1960, a Dahlgren team led by Dr. Cohen and Mr. Richard Anderle had developed highly accurate orbit determination capabilities in support of the TRANSIT navigation system, a satellite system developed to provide navigational updates for POLARIS submarines. Dr. Cohen and Mr. Anderle developed techniques to deduce the mathematical representation of the earth's gravity field using *dynamic geodesy;* i.e., by using observations of the motion of satellites in orbit. After only one month of Doppler observations on the Transit 1B satellite, Dr. Cohen and Mr. Anderle were able to verify the *pear-shaped earth* gravity field and reported it in *Science* magazine in 1960.

At about the same time, this team also codeveloped, with the Naval Research Laboratory, the Navy Space Surveillance System for determining orbits and categorizing objects orbiting around the earth. Through its work in these two programs, the Dahlgren team developed into national leaders in accurately determining satellite orbits, which continues to this day. Dahlgren pioneered satellite geodesy to improve the knowledge of the earth's gravity field and reference system, developing World Geodetic Systems (WGS) 62, 66, and 72, and made important contributions to WGS 84. Dynamic geodesy was inadequate to model the higher frequency terms of the gravity field. To solve this problem, Dahlgren performed mathematical analyses that became the basis of *satellite altimetry,* employing radar measurements of a satellite's altitude above the sea surface. The use of satellite altimetry, together with dynamic geodesy, permitted inference of the geoid, or mean sea level, and the earth's gravity field, including high-frequency components.

The team was also the technology leader in developing Doppler point positioning. The computational capabilities and technology developed at Dahlgren for determining the precise ephemeris of navigational satellites, and for point positioning, were transferred to the Defense Mapping Agency in 1975. The first studies for the Global Positioning System (GPS) constellation were conducted between

1969 and 1972, leading to the production of precise ephemerides for GPS—a high-accuracy, worldwide, space-based navigation system. Dahlgren's contributions and innovations related to GPS have been many and continue. Some of these include point positioning, or surveying capability, man-portable receivers, and determining attitude and orientation as well as position.[6]

Other Contributions with Wide Impact

Over the years, the breadth of significant mathematical and computational technology contributions made at the Dahlgren laboratory have impacted many fields of study within and outside DoD. Space limitations here prohibit a full discussion of Dahlgren's many noteworthy achievements over the years. However, three of these accomplishments that have resulted in widespread use of Dahlgren products are briefly described below.

Developing an accurate model of the world's ocean tides is a problem that has been of great interest for over 300 years, as evident by the more than 5,000 papers that have been published on the subject. The interest was primarily academic at first, but in the twentieth century, military and civilian applications in oceanography, geophysics, and meteorology have raised the interest level. At Dahlgren, the interest was driven by requirements to support gravity earth figure modeling for missile trajectory and satellite ephemeris computations. During the 1970s and 1980s, Dr. Ernst W. Swiderski attacked this problem and, not only produced the first usable model, but extended it to include all of the principal tidal constituents, achieving the astonishing overall accuracy of 10 centimeters for the open oceans. His publications, data, and computer programs have been requested by hundreds of scientists across the world, and his model has been accepted as the standard by more than 20 national and international institutions.

In 1976, a project was undertaken at Dahlgren to develop a general purpose mathematical software library to be made available to the entire Naval Surface Weapons Center (NSWC) scientific community. The objective was to develop mathematical algorithms and implement them as computer subroutines in order to provide techniques to users who may not have had the expertise or the time and money to develop them for themselves. It was also intended to prevent the loss of productivity resulting from redundant development of subroutines by many users across the Center. Alfred H. Morris was given responsibility for the project, and due to his efforts, supplemented with contributions from other mathematicians and research scientists (especially A. V. Hershey and A. R. DiDonato), a very successful library has been developed. It contains routines that permit handling problems which would otherwise be difficult to solve without their availability, and for which no other known solution implementations exist. As the library developed, it became clear that it would be of value throughout the U.S. defense community and beyond, and was made available, without charge, to those who expressed a need. The library has become very popular, and hundreds of copies have been distributed by NSWC to sites across the U.S. and the world. Furthermore, these sites have, in turn, expanded distribution of the library (for example, one user in Australia indicated that he had distributed more than 50 copies in the previous six months). It seems safe to say that thousands of users around the world are now benefiting from this library.

Another set of products that has been widely used relate to the research and implementation that Dr. Allen V. Hershey achieved in cartography and in typography. His mathematical representations of maps and fonts became the basis of the entire product lines of several new companies formed in the early days of personal computers and are still in wide use throughout the world today.

Conclusion

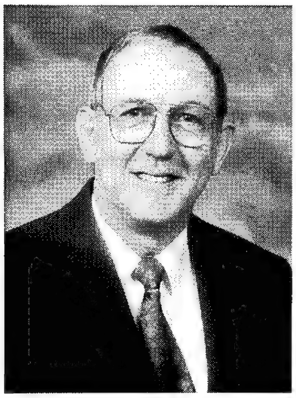
The Dahlgren laboratory persists in anticipating new requirements related to the strategic and space systems programs for which it has long been an innovator. Development of new mathematical techniques, representation of improved theories, and computer implementations continue to move forward. The knowledge gained and the successes achieved from these

efforts have contributed fundamentally towards Dahlgren's ability to obtain vital responsibilities in AEGIS and TOMAHAWK. Dahlgren is prepared to make similar contributions to new Navy systems and requirements. NSWCDD mathematicians and scientists continually pioneer new areas in mathematics and computer science, making both basic research extensions to those fields and innovative implementations of the new technology. This includes such areas as artificial neural networks basic research, with investigations of pattern recognition applications to problems as varied as weapons target identification and x-ray mammography analysis. Dahlgren continues to provide leadership in mathematics and computing technology for the next generation.

References

1. McCollum, Kenneth G. (editor), *Dahlgren*, Naval Surface Weapons Center, Jun 1977.
2. Brown, David R., Jr., *The Navy's High Road to Geoballistics, Part I*, Naval Surface Weapons Center, Dec 1986.
3. Niemann, Ralph A., *Dahlgren's Participation in the Development of Computer Technology*, Naval Surface Weapons Center, Sep 1982.
4. Dongarra, Jack J., *Performance of Various Computers Using Standard Linear Equation Software*, University of Tennessee, 22 Jul 1993.
5. Edwards, Donald T. and Daniels, Doreen H., *The SLBM Program: Brief History of NSWC's Involvement in the FBM/SLBM Fire Control Systems, 1956-1984*. Naval Surface Weapons Center, 1984.
6. Hughey, Raymond H. and Hill, Robert W., *Strategic and Space Systems*, Naval Surface Weapons Center, Aug 1985.

The Author



RAYMOND H. HUGHEY, JR., was employed at the Dahlgren laboratory for almost 35 years from 1959 until his retirement in 1994. He was a Senior Scientist in the Strategic and Space Systems Department from 1982 to 1994. He also served as Head, Electronic Warfare Division, from 1979 to 1982 and Deputy Department Head, Strategic Systems Department, from 1978 to 1979. He played various roles in the SLBM program from 1959 to 1978 including Head of the FBM Geoballistics Division from 1967 to 1978. Mr. Hughey earned a B.S. degree in Physics from the University of Alabama in 1958 and completed graduate work in physics and math at the University of Kentucky, the American University, and George Washington University. He is a graduate of the Federal Executive Institute (1977), a recipient of the John Adolphus Dahlgren Award, and currently works as a part-time consultant on Navy systems matters.

Evolution of Combat System Computing

Daniel T. Green

This paper discusses the evolving use of digital computer technology in U.S. Navy surface ship combat systems. It presents some of the underlying characteristics of both computer technology and combat systems that impact this evolution. This paper addresses primarily the use of computers in real-time weapon and combat control.

Introduction

The title of this paper as stated is hopelessly ambitious, even if implicitly limited to the role of computing in U.S. Navy surface combat systems. Reference 1 defines a system as "an assemblage of objects united by some form of regular interaction or interdependence; an organic or organized whole." This definition of system is still correct and applicable. In a broad sense, a Navy combat system is a group of ships acting in concert to achieve the desired military objective. In the view taken in this paper, the Navy combat system is the ship as a whole, including its weapons and people. From this view, computing in U.S. Navy combat systems clearly started with the first Navy ship. From earliest times, computation has always been required—not only during combat evolutions, but also in the design stages of a ship and its weapons. A complete study of the evolution of combat system computing would have to cover its role from the beginning of the Navy up to the present, its role in ship and weapon system design, and its role in weapon system employment. My goal in this paper is much less ambitious—it is to follow the use of digital computers as a component of surface Navy deployed combat system elements. I will touch upon the role played by the Naval Surface Warfare Center, Dahlgren Division (NSWCDD) throughout its history in this use from 1958 to 1995, my tenure at the Dahlgren Laboratory.

The ideas presented here in this article are based on my involvement in this area and on a review of the references cited at the end of this article.[1-5] In reviewing my experience and the references, I came to the following simple, yet to me, critical observation. Hardware, software, and design tools available to the combat system or subsystem designer and implementer have improved dramatically over the period of time covered here. However, the underlying problems to be solved and the basic design approach to their solution have remained essentially unchanged from the initial use of digital computers in surface ship applications in the 1960s. The pioneers in this field showed a remarkable insight into the essential nature of the system design problem and the most appropriate approach to its solution.

The name of the Dahlgren organization evolved over this period from Naval Proving Ground to Naval Weapons Laboratory to Naval Surface Weapons Center to Naval Surface Warfare Center to NSWCDD. The name changes track the ever-increasing integration of combat system components into larger system complexes in Navy weapon system and combat system design. This set of organizational name changes reflects the organization's evolution from attention to individual components (large-caliber guns, armor, and projectiles) to the broader aspects of Navy combat systems (Fleet

Ballistic Missile (FBM) program, AEGIS, TOMAHAWK, Ship Self-Defense System (SSDS), and Theater Air Defense (TAD)). It remains essential that each individual component of a combat system be well designed and implemented so that it may carry out its assigned functions properly and in concert with all other components. The need to pay ever-increasing attention to the combat system (ship or fleet) as a unit is driven by the increasing complexity of warfare, reduced reaction time to threats, and increased automation and integration of various weapons and sensors so as to reduce manning requirements and improve system responsiveness. Computer technology is a critical tool in achieving the desired levels of automation and integration.

Computer Technology Characteristics

The computer technology of interest here has a number of important components. The computer hardware is the component usually first considered. This includes the central processing unit (CPU) and associated memory; methods of interconnecting computers (backplanes, point-to-point channels, and networks); and computer peripherals, including mass storage devices. These peripherals range from A-D and D-A converters to sophisticated printers and graphic display devices. This hardware, however, is generally of little use without some form of:

- Operating system or executive to control the hardware components and programs
- Languages in which to express our directions to the computer hardware (program)
- Effective procedures to design programs
- Methods (mathematical or otherwise) to carry out the desired functions on the computer and communication hardware

Where these functions have general applicability, they are frequently packaged into units that can be used either as part of a larger program or independently. The first attempts at such packaging were subroutines to carry out mathematical functions or communicate with computer peripheral components. Today this simple beginning has expanded to include spreadsheet programs, word processors, and sets of communication protocols. Today, in many cases, cooperative organizations exist (e.g., International Standards Organization (ISO), American National Standards Institute (ANSI), and Open Software Foundation) to define standards for these functions. These standards represent agreements by vendors and users so that products from different sources can interoperate, be portable, and/or present the same interface to the user.

Selecting from among multiple ways of performing the same operation is a problem that is an integral part of the rapidly evolving area of computer technology. Since we are continuously expanding the scope of applications and learning new and, hopefully, better ways of doing things, it takes time and effort to identify and select an agreed-to way from among a set of similar solutions. For example, in 1958 NPG had two sets of mathematical subroutines for use on its computer, the Naval Ordnance Research Calculator (NORC). One had been generated by the set of programmers under the leadership of Mr. John Walker, the other by Dr. Alan Hershey for his use. As computer technology has evolved, this situation continues wherein multiple sources are capable of implementing required functions. Generally, now there are multiple competing variations on ways to implement any computer function.

This gives rise to a major problem in the design of large and long-lived programs such as those that are a part of combat systems. Typically such systems contain parts (subsystems) designed and implemented by different groups of people at different times. If these parts are to work together (interoperate) and be reused over time or in different subsystems (portability), then standards must exist and be enforced at least for the interfaces between components. The selection of such standards without unduly constraining improvements in functionality is a major issue for the combat system designer. This tension among supporting continuing system evolution, allowing sufficient flexibility for subsystem implementers to do their job, and providing interoperability/portability has existed since the beginning of digital computer technology, as the example of Mr. Walker and Dr. Hershey illustrates. The Navy and, more broadly, the Department of Defense (DoD), have over the years approached this problem in a variety of ways. An approach of interest here is the generation of Military Standards and other directive documents that require

or suggest the use of development procedures, minimum performance standards, specific end items (computers, languages), and interface standards. Only the last two items will be touched upon in this article.

In a general way, the evolution of using digital computer technology in Navy combat systems follows the same trend as in our society as a whole. The earliest applications were to assist in carrying out functions in individual weapon system or subsystem components. The integration and coordination of the various weapon systems into the ship and fleet combat system was supplied by the people and their operating procedures. Since 1958 two important trends have continued. First, our understanding of how to use digital computer technology to assist or automate activities formerly dependent upon the human intellect is growing rapidly, perhaps exponentially. Second, digital computers and associated digital data transmission capability is becoming ever more capable and cheaper. Thus the use of digital technology has spread both to take over more and more functions of weapon system components and to assist in automating functions and communications previously carried out by people. Thus it is now possible to use commercial products to merge multiple computers into a single system, whereas at the beginning of the period under discussion, we could barely control the integration of specially designed computers with a small set of components.

Role of Computer Technology in Combat Systems

From their first use in the 1960s to today, digital computers, data transfer components, and the software used in them have been considered integral and essential components of the weapon or combat system. However, they are only a small part of the physical plant and an even smaller part of a completely useful Navy weapon system. Reference 1 serves as a useful starting point to put computer technology into its proper context as it states:

The complete weapons system does not consist of the physical equipment alone, but also of technical manuals, training plans and equipment, and spare parts, and an optimum employment doctrine. Weapon system development thus includes concurrent development of all these items. Planning, funding, and development must include those elements which must be completed by the time the weapon is ready to be introduced to the fleet.

In that context and at that time, the computer and its software were considered part of the physical equipment. Figure 1 illustrates this definition of the necessary parts of a weapon system circa 1960. The increasing use of digital computer technology has a significant impact on each of these areas mentioned, either directly or indirectly. Initially, the direct impact was on the physical equipment. Analog equipment was replaced by digital computers and software. New functions were implemented in digital computers. As this progressed, concurrent changes were required in the other areas. In particular, training plans and equipment needed to be modified. New or improved capabilities required changes in employment doctrine. Maintenance plans needed to consider the peculiar nature of software for digital computers. Changes in software alone could change the performance characteristics of

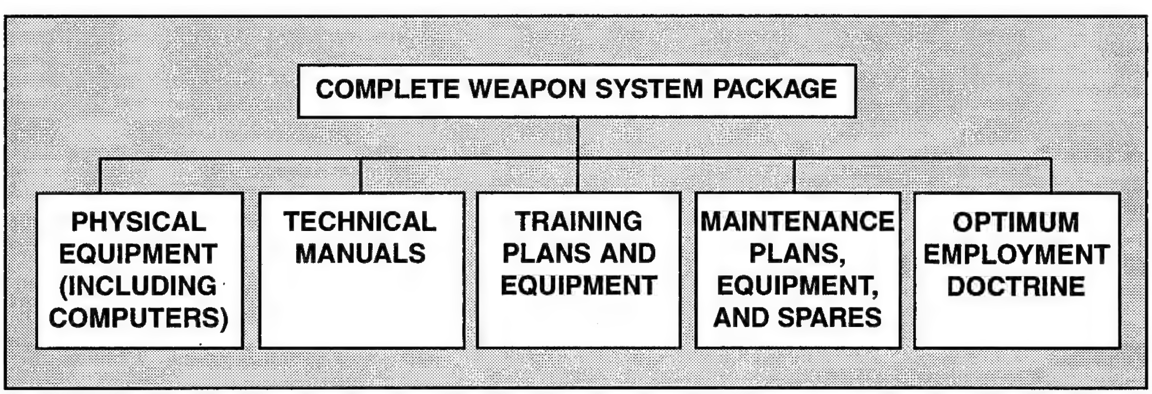


Figure 1. *The complete weapon system package (1968).*

the system. The wear and failure characteristics of software are dramatically different than those of hardware. New levels of complexity were introduced into the system.

Today, not only do these indirect effects impact areas other than physical equipment, but computer technology is used to carry out the functions of these areas. For example, there is a growing trend to store all technical manuals and related documentation digitally. It is hoped that, in the future, paper will be the exception. Digital computers, via simulation programs and other decision aids, play an essential role in generating optimum employment doctrine. This may be done offline or dynamically during a battle.

Reference 1 also notes that the best weapon is the most economical one to design, build, and use that will effectively achieve the desired military objective. Objectives in weapon system design are listed in Reference 1 as:

- Effectiveness/performance
- Operability
- Reliability
- Versatility (capable of a wide variety of missions and adaptable to future requirements)
- Invulnerability (includes resistance to countermeasures, hostile physical environment, and battle damage)

These underlying principles are equally applicable to combat systems designed today and must still control the choice of computer technology to be used in weapon and combat systems.

However, advances in computer technology as well as the growing need to integrate more closely and automate the integration of individual weapons into a combat system have significantly changed the definition of a complete combat system. In simpler times, the integration of weapons into a combat system was achieved by their physical placement on the ship and by the doctrine implemented by the ship's crew concerning their use together. Reference 1 recognized that this was in a state of flux as it states:

> *In the recent evolution of weapon technology, one of the most striking developments has been the transformation from the single weapon to the complex weapon system. Such a system is made up of a number of unique, specialized components which must be coordinated to achieve overall effectiveness.*

Components listed are generalized in an accompanying figure as control, detection, tracking, launching, designation, direction, and stabilization. The major factors that must be considered in the design and development of weapon systems are listed as system requirements, sensors, computers, communications (both device-device and human-machine), and system dynamics.

As we evolve from the 1960s' to today's practice, it is my belief that a complete combat system is better represented by Figure 2. The additions to Figure 1 are in bold type. The obvious change in this figure is the addition of a formal combat system structure plan, which provides both control and information flow structures. This structure is carried out by the provision of standards and policies that control both the implementation process and the actual operation of the final set of components to be merged into a complete combat system. This structure is intended to allow for both the interoperability and portability of system components while meeting the objectives of weapon system design noted above. Because of their importance in current designs, digital computers and accompanying software are now explicit parts of the subsystem package. This package must also make explicit provisions for how it is to be integrated into the complete system and perhaps into a variety of different combat systems.

Less apparent is the migration of computer technology into all other components of this package. Technical manuals are being digitized (frequently generated by computerized word processors and computer-aided design (CAD) systems) and stored in a digital format for use on the ship. Training is now computer-assisted by simulation systems. Shipboard training on the actual system is now typically incorporated into the weapon system design. Maintenance is computer-assisted by automated diagnostics that will isolate failures to well-defined small components (line replaceable units). In many cases, and increasingly in the future, built-in fault tolerance capabilities will automatically bypass the use of failed components. As

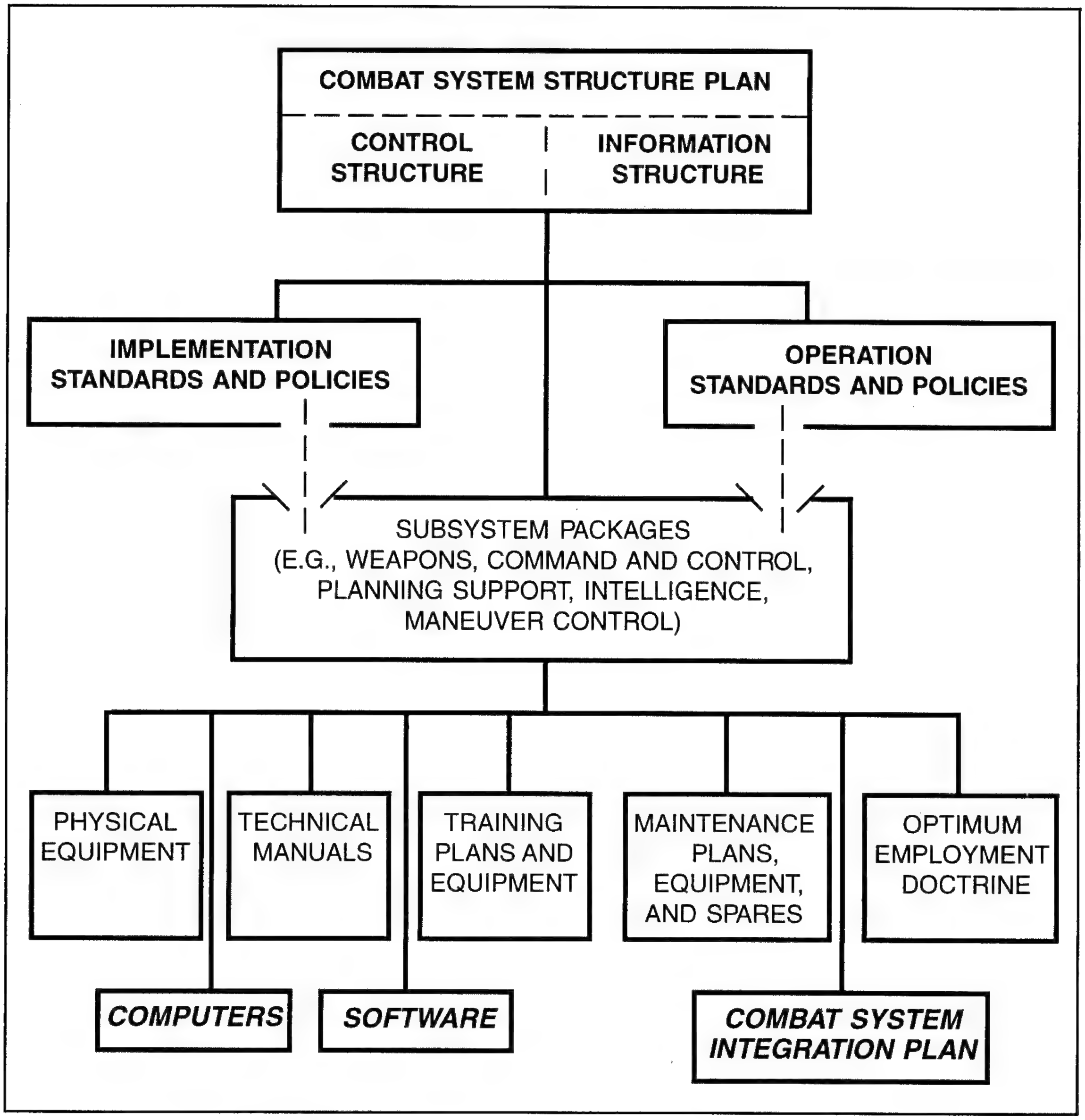


Figure 2. *The complete combat system package (1995).*

computer-supported planning aids become even more pervasive on ships, they can be used to help evolve the employment doctrine to meet new challenges and will continue to be adapted to the particular skills and preferences of the crew.

Combat System Computer Architecture

The earliest use of a general purpose digital computer as a component of a Navy combat system was in the Navy Tactical Data System (NTDS). According to Reference 2, the first test of the digital computer for use in this system was in 1958. This computer was a solid-state machine using discrete transistors. At this stage, a basic design principle for Navy combat use was recognized. This approach is summarized in the following two quotes from Reference 2.

A building block concept would be employed, using standard computers, displays, and communication terminals to meet the varying requirements of different ship types and inevitable changes in sensors, weapon systems, and tactics which occur over the life of a given ship type.

Variable sizes of computers consisting of different mixes of a standard set of functional modules were first considered. Then Dr. Joe Eachus, a U.S. Government com-

puter scientist and an early advocate of general purpose computers, suggested a "unit computer" approach. The idea of standard size, stand alone, independent computers, all interconnected and working on different parts of the same problem was completely new. It had never been tried . . . .

The basic idea of a number of units cooperating to achieve the combat system objective was further developed in Reference 1 in the section on digital computers. It is interesting to note that at that time electronic analog computers were recommended for some computing applications because of their speed. Thus the term "hybrid computer" to refer to a combination of analog and general purpose digital computers is used in the following quote. The figure is copied from Reference 1.

A hybrid computing system as it might occur in a complex fire control situation is illustrated (see Figure 3). *The tracking, prediction, and weapon control (launcher bearing, guidance, etc.) calculations are performed by digital computers, while the search control, detection, display, coordinate transformation, and target evaluation and weapon assignment calculations are performed by analog computers or men. This combination makes use of the multiplex ability of digital computers, in order to handle many targets and many weapons simultaneously, while accommodating the analog nature of man. The essential trigonometric calculations of searching and coordinate transformation are handled by analog methods also. Note that the system requires two analog-to-digital converters.*

In 1995 terminology, a peer-to-peer, distributed computing approach was being advocated in these early designs. The major design difference in a current design is that most of the analog computers are now replaced by functions carried out on digital computers, with the A-D and D-A converters being moved closer to the endpoints of the system. The data transfer among the devices will be via digital connections (e.g., backplanes, shared memory, local area networks).

In these first attempts, the designers and implementers of weapon systems had to generate their own methods and tools. This included the design of the computers, interconnect mechanisms, and display devices. This led to a series of computer families and display families, specialized languages, operating systems, and other support software.

It is convenient to divide combat system computation into three areas—its use in

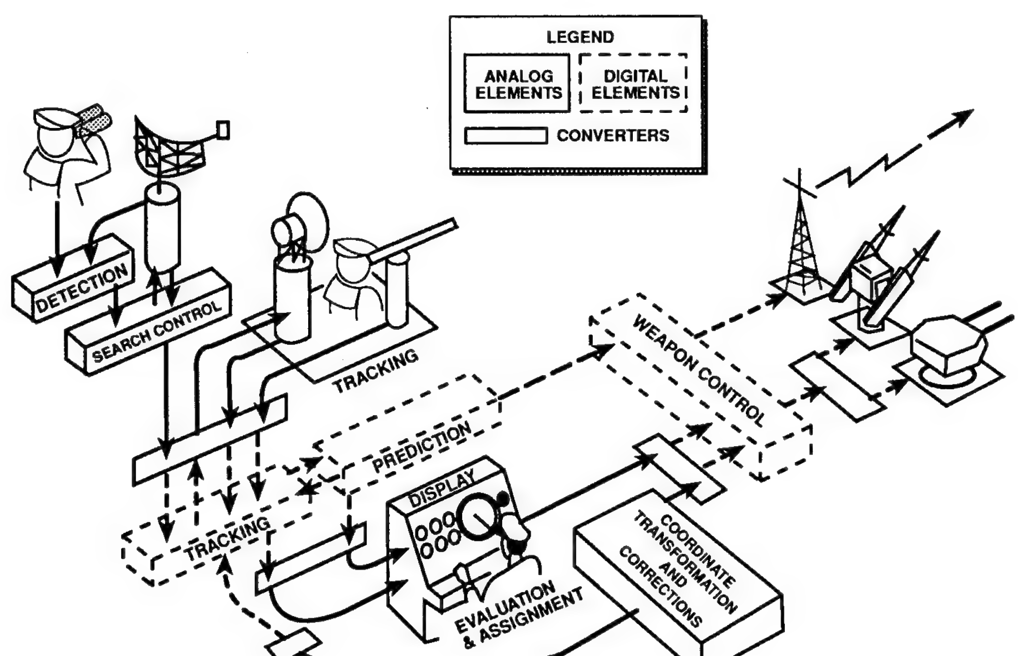


Figure 3. *Computing system design, circa 1968.*

weapon system and combat system control; signal processors; planning functions, decision aids, or other support functions. Each of these applications places different burdens on the hardware and software. This article is centered on the first, as applied to weapon and combat system control. It will give limited attention to the second and third applications.

Weapon System and Combat System Control

The NTDS and weapon control programs mentioned above belong to this first application. Weapon and combat control computer programs have had requirements that differed significantly from other early uses of computers. This arose from the need to keep track of events (e.g., own ship position and motion, target position and motion) occurring in the surrounding physical environment, and to control equipment (missile launchers, guns, radar directors) in response to these events. From the first, the program was broken up into a number of cooperating modules or processes, each of which carried out some particular function. Certain of these processes were triggered periodically, others in response to specific events in the environment. All had to complete their work within a specified time limit so as to maintain the internal or computer view of the world consistent with reality. In current terminology, they had to operate in real time. Ideally, these programs operate correctly and without interruption for the months-long duration of the ship's mission. The first operating systems were very simple executives that carried out the scheduling of the periodic functions and handled the interrupts or other mechanisms that signaled the aperiodic events. As the systems grew more complex, more functions were shifted to these executives so that they became full-blown, multi-CPU, multiprocess, distributed (if very specialized) operating systems.

The AEGIS Automated Tactical Environmental System (ATES) executive system illustrated in Figure 4 marks an advanced stage in this

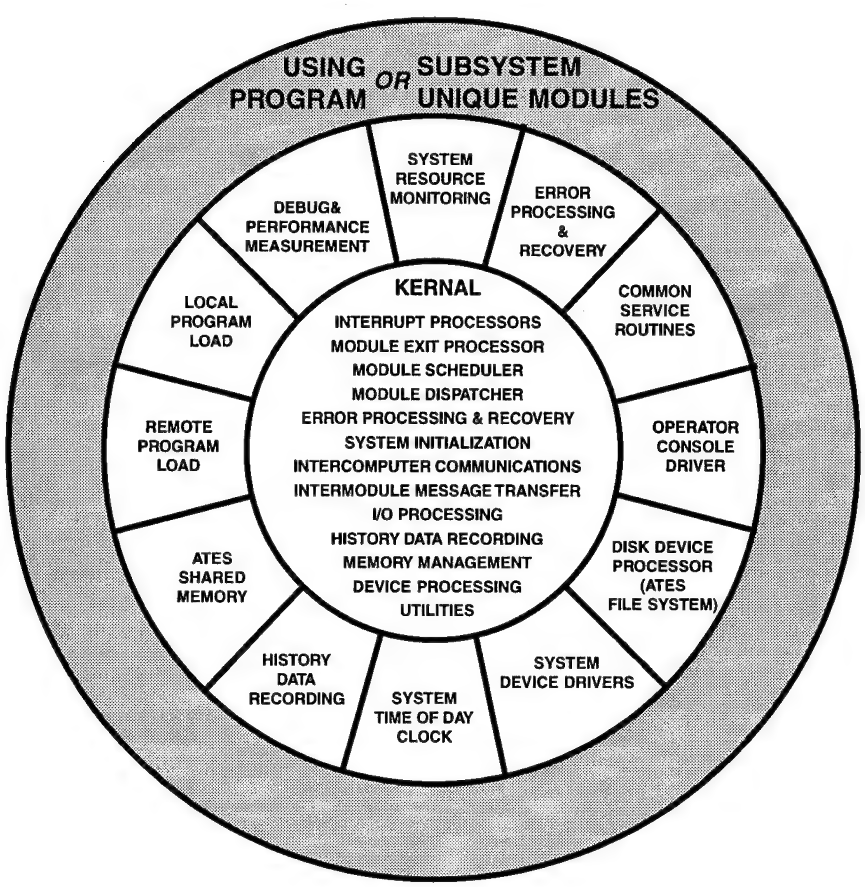


Figure 4. *AEGIS Tactical Executive System (ATES/43) services.*

evolution. In today's world, there are commercial applications, such as automated factories, that have similar problems. This results in some products being available in the marketplace and in standards, such as the Institute of Electrical and Electronics Engineers (IEEE) operating system standard, profiles for "real-time portable operating system interface (POSIX)" and the ISO Manufacturing Message Specification network standard, which may be useful in future combat applications. While today there are more tools available to help the program designer and implementer of control system programs for combat systems, the underlying problem and approach to its solution remain unchanged from the 1960s. In addition to operating system tools, another tool of particular interest is the DoD-sponsored language Ada. One of the features of this language is the task construct. This is a formal way of expressing in a high-order language the processes, scheduling, and intercommunication required in control programs.

Another control operation that will need to be integrated into the combat system of the future is ship machinery, propulsion, and maneuvering control. Automation is already in use in this area.

Signal Processing

The second application of digital computing technology in combat systems is its use in signal processing. This involves extracting, in real time, information from the signals received by devices such as radars, sonars, passive electronic receivers, and optical devices (e.g., television cameras). The problem here is to extract the interesting information (targets) from the mass of information provided by these sensing devices. At the beginning of this period, this was typically performed by a person viewing the information on some form of a display device. As understanding of the physical and intellectual processes involved improved, more of the detection and analysis was assigned to specialized digital computers running algorithms developed for the process. These computers tend to consist of a set of processors interconnected in a way suited for the problem at hand. In some cases (e.g., radar), the process is now highly automated. In other cases, considerable human involvement still remains. As is shown in Figure 3, the extracted information is provided in a digital form to other combat system elements for processing.

Planning Support

The third application that introduced computer technology to combat systems was through their use in planning support or as decision aids. Two early examples of this use developed at Dahlgren were a mine warfare simulation program, which in 1970 was reprogrammed from the base's commercial computers for use on UNIVAC 1500 fleet computers, and an automated "frequency assignment procedure" for the TERRIER, TARTAR, and TALOS (3 Ts) missiles which was generated in 1971 and programmed for use on the 642 class of computers. This latter was an automation of an existing manual procedure that ensured that the frequencies assigned to individual missiles and radars were such that mutual interference was eliminated. As commercial computers became smaller, less expensive, and easier to use, the generation of such tools for use on combat ships in a support role became common both by shore support personnel and by ship personnel. These tools led to the development of formally fielded and supported systems such as the Naval Tactical Command System-Afloat (NTCS-A). Such tools were fielded early on commercial minicomputers and later on personal computers (PCs) and workstations. These tools made use of common operating systems and languages such as DOS and BASIC and incorporated other commercial software such as database management systems.

While some planning systems are interfaced with combat control and fire control systems, they do not share their real-time nature. However, as various weapon and support systems are integrated to form more effective and less manpower-intensive combat systems, ways to allow these disparate components to work together will have to be improved. Since some of these support operations require the use of highly classified data,

the problem of providing data security and protecting the integrity of weapon control operations in a unified system will become increasingly important and demanding. The TOMAHAWK weapon system already marries a planning and weapon control function into a single system.

Combat System Computer Technology and Commercial Technology

Table 1 lists some of the major computer and display components available for use since 1958 in combat systems. Design and availability dates are given when known. This list is incomplete in that other similar devices were also used during this time period. At first both computers and display units had to be designed specifically to meet Navy combat system requirements. Early commercial solid-state digital computers of this period with sufficient computational capacity (e.g., IBM 7090) were too big (a large roomful of equipment), would not survive in the physical environment of a Navy ship, and had an instruction set and memory design (instruction set architecture or ISA) that lacked features required in combat and fire control applications. The primitive operating systems and the support software that accompanied them was also unsuitable for the development and fielding of weapon control and fire control systems, having been designed for other application areas. The display devices available were unsuited to the input/output needs of fire control and combat control watchstations.

As a result, the Navy first designed the 642 class of computers and UYA 4 family of console devices (see Reference 2). These were followed by the AN/UYK 7 (32-bit ISA) and AN/UYK 20 (16-bit ISA) families of computers and, at a somewhat later date, by the UYQ 21 family of display devices. The UYK 43 was designed to provide an enhanced but upwards compatible version of the UYK 7 ISA, and the UYK 44 similarly followed the UYK 20. Each succeeding family of computers dramatically improved performance and allowed much more computing capability to be provided within the same or smaller space, weight, and electrical power budget than that of the previous generation.

These devices were declared as standards by the Navy for use in all combat control and fire control applications. During this period of time, this approach was, to a large extent, unavoidable since commercial products were, by and large, functionally as well as physically unsuitable for application in combat systems. Usually these standard families were accepted. The exceptions tended to use ruggedized or militarized versions of commercial mini-computers, which were becoming available in the 1970s. The use of standard Navy-designed components eased many problems associated with training and logistic support but meant that the Navy was out of the mainstream of computer hardware and software development in

Table 1. *Computer and console equipment*

COMPONENT FAMILY	DESIGN DATE	FIRST AVAILABILITY	PLANNED USE
CP-642	~1955	1958	CONTROL COMPUTER
UYA 4	1956	~1960	DISPLAY CONSOLE/WATCHSTATION
AN/UYK-7	~1969		CONTROL COMPUTER
AN/UYK-20	~1973		CONTROL COMPUTER
UYQ 21			DISPLAY CONSOLE/WATCHSTATION
AN/UYK43	~1980	~1986	CONTROL COMPUTER
AN/UYK44	~1980	~1984	CONTROL COMPUTER
TAC 3		1992	PLANNING (WORKSTATION)
AN/UYQ-70	~1994	1995	CONSOLE & CONTROL COMPUTER
TAC 4		1995	PLANNING (WORKSTATION) & CONTROL

its combat computers. It also provided the Navy with complete control over both the hardware and software, and the upgrades thereof, avoiding many concerns with the reliability, safety, and security of fielded systems.

However, advances in integrated circuitry, which make computers ever smaller and ever cheaper, and the widespread introduction of computers into many commercial fields of endeavor have dramatically changed the situation. In the 1960s and 1970s, memory, CPU capacity, and communication bandwidth among interconnected devices were all limiting factors. This has changed over time so that, in many applications, this is no longer true. The limiting factor now is generally our capability to understand the application area so that it can be automated or assisted by the use of computers. Commercial applications similar in many ways to combat and fire control applications are evolving. The physical size of powerful computers has shrunk to the point where a variety of packaging techniques are possible that will allow commercial products with relatively minor modifications to survive in the shipboard physical environment.

As a result of this (as well as congressional direction), the Navy is now taking a different approach in introducing computer technology into combat systems. In current computer jargon, an Open-System Architecture (OSA) approach is being used. An early instance of this new approach is the Tactical Advanced Computer (TAC) series of competitive procurements by Space and Naval Warfare Systems Command (SPAWARSYSCON). This computer and display procurement approach did not define an ISA but was aimed at procuring commercial equipment that had standard/open interfaces defined so that the contract could be periodically recompeted with the possible selection of different equipment on which the older application software could run. At first, these procurements were intended only to supply computer technology to be used in nonmission-critical applications, such as planning or off-line decision support systems. As progression is made to the TAC 4 contract, and as, with more integration, the distinction between planning/decision support and combat control blurs, the TAC 4 family may be used in some control applications. Following current commercial practice, the computer and display capabilities are now integrated into one logical entity called a workstation so that the distinction between computers and consoles is also now being erased.

The AN/UYQ-70 (also known as the Advanced Display System (ADS)) shows another path to introducing commercially based systems into the combat system. It is the result of a competitive contract that represents Naval Sea Systems Command's (NAVSEA's) first acquisition of a mission-critical system using an OSA integrating commercial off-the-shelf (COTS) technology and nondevelopmental items (NDI). It is a functional follow-on to the families of display consoles or watchstations that started with the UYA 4 used in the first NTDS systems. Using currently available computer technology, this family of devices also has the capability to carry out the computational functions formerly supported by computers of the UYK 7, 43, 20, and 44 classes. It includes not only computer and display hardware but also operating system and other support software as well as networking capability. While the equipment and packaging of this family is different from the TAC 4, they share many functional specifications and even some common components. Figure 5 illustrates some of the packaging changes made possible by advancing computer technology. On the left is a picture of a prototype AN/UYQ-70 console. On the lower right is a picture of a four processor UYK-7 computer, and on the upper right is an OJ451 console of the UYQ-21 family. Because the consoles must be used by people, they have approximately the same size and shape. However, the AN/UYQ-70 can house in its cabinet both an improved display capability and more computational capability than contained in the UYK-7 computer. Thus, we can now package both display and computational capability in a single cabinet rather than separating them. This simple change potentially has considerable impact on system design.

This new procurement approach offers the promise of making current state-of-the-practice computer technology more readily available in combat systems. If properly managed, this

should reduce cost and improve system performance. However, it is not without its problems. The potential rapid upgrading of components offers many interesting logistic and training problems. It appears likely that a very different set of solutions than those currently used will need to be found. A second issue centers around the complexity of modern standards, their many options, and the difficulty of ensuring that products comply with them. Even given that vendors will be making their best effort to comply with open standards, there is no guarantee that unexpected results will not occur when one upgrades from a TAC N to a TAC N+1 line of components or to the next vendor's version of the ADS. Finally, the generality and increased complexity of these commercial products, as well as their frequent change for reasons outside Navy control, will make far more difficult the problem of ensuring that predictable performance, safety, reliability, and security needs are met. The only known way to evaluate these characteristics of a computer system is via testing. Yet, given the nonlinear behavior of a digital system, especially complex ones, even extensive testing will not uncover all the faults in a system. One of the problems is that very small changes in input or the state of the system may give rise to very large changes in output and system behavior.

Summary

Computer technology has evolved dramatically since 1958. This includes hardware, software, and our ability to use computers to assist in intellectual processes. While many aspects of computer technology are still in the research domain, many others have evolved to readily available commodities. This implies that our techniques of applying computers to combat system problems must change to reflect these

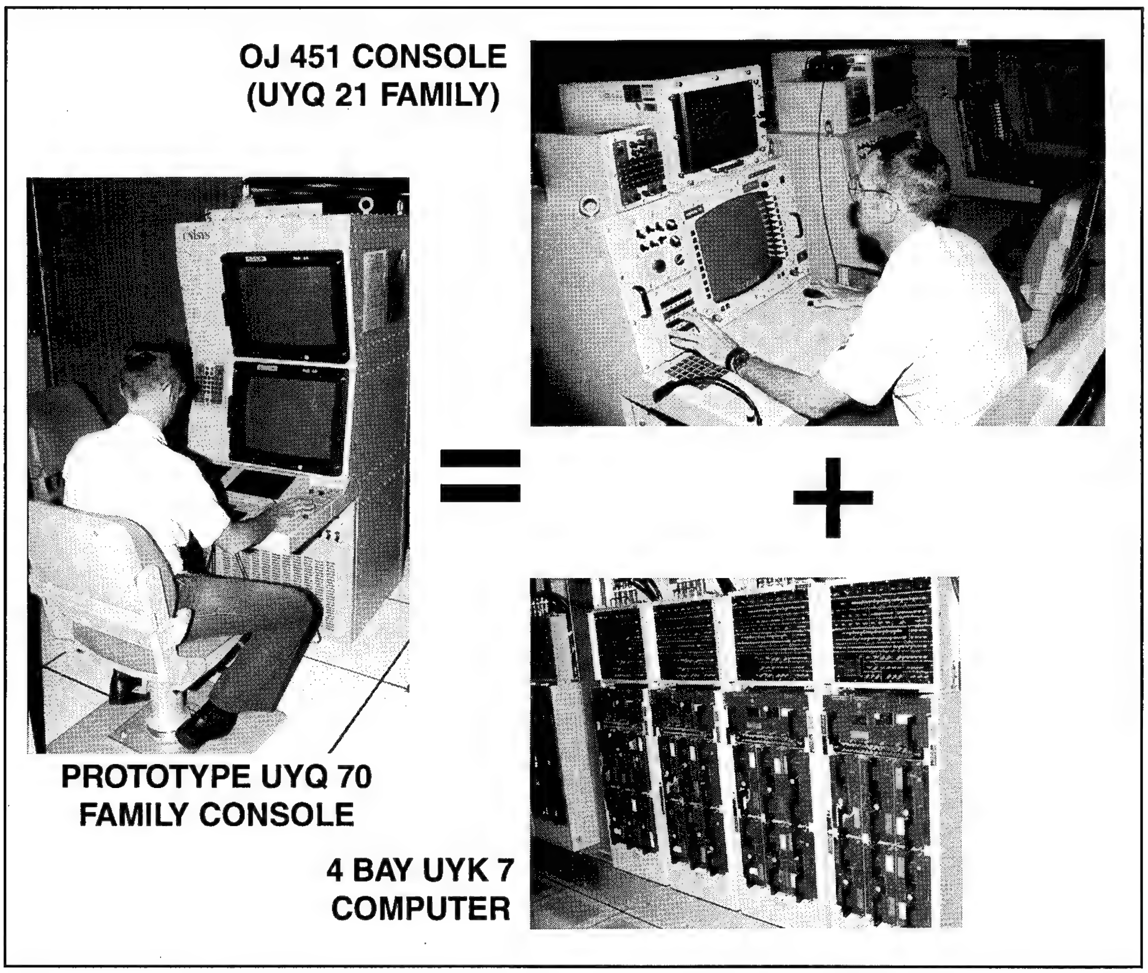


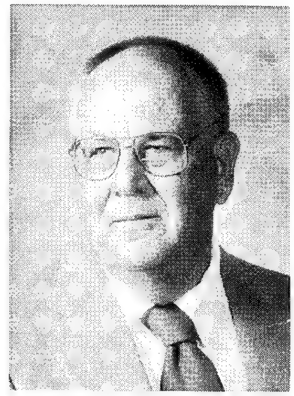
Figure 5. *Changing computer technology's impact on device packaging.*

changing realities. However, the basic problem and basic approaches used in combat system computing have not changed. Our understanding of the techniques involved has evolved and improved. More and better tools are available with which to achieve Navy objectives. However, the ever-growing complexity of increasing automation and integration in combat systems maintains the level of difficulty of the work to be done by those implementing systems for Navy use.

References

1. *Weapon System Fundamentals, Synthesis of Systems*, NAVORD OP 3000, Vol. 3, Government Printing Office, 1968.
2. Swenson, Mahinske, and Stoutenburgh, "NTDS—A Page in Naval History," *Naval Engineers Journal*, May 1988.
3. *Department of Defense Computer Technology, A Report To Congress*, Office of the Undersecretary of Defense Research and Engineering, Aug 1983.
4. Niemann, Ralph A., *Dahlgren's Participation In The Development of Computer Technology*, NSWC MP 81-416, NSWC, Dahlgren, VA.
5. *Command History: Naval Weapons Laboratory 1959-1972, Naval Surface Weapons Center 1974-1987.*

The Author



DANIEL T. GREEN completed his duty at NSWCDD as a staff scientist in the Combat Systems Technology Group of the Systems Research and Technology Department. He holds a B.A. in mathematics from Niagara University and an M.A. in mathematics from Catholic University. He has taken advanced work in mathematics at American University, in operations research from Case Institute Of Technology, and in management from Oklahoma University. He was employed at NSWCDD from April 1958 to September 1995. During this time he worked as a computer programmer, computer systems analyst, and computer systems consultant. During this period, he headed five divisions at NSWCDD and was a plank holder in two departments. His latest tasks revolve around developing information network technology for incorporation in future combat systems.

An Overview of Total Ship System Engineering–A "Gang of Six" Initiative

James R. Pollard and Bernard G. Duren

This article considers an effort to reinvent the process by which the Navy transforms operational requirements into surface combatants. The work is being conducted as a collaborative effort involving a group of key Navy acquisition executives and three warfare centers. The strategy is to articulate a new framework for total ship system engineering. For this, three things are necessary:

- *Open and continuous dialog on war-fighting requirements*
- *A family of backbone control structures that provide the means for mission teams to operate a ship as a coherent entity*
- *Agreed-to concepts, standards, and tools for design integration*

Subsequent efforts will address how the approach can be implemented in future programs.

"The Gang"

The activities listed in Figure 1 are the main contributors, although others have participated at times. The flag-level members are supported by an administrative team drawing from their staffs as well as the Naval Surface Warfare Center (NSWC) and Naval Command, Control and Ocean Surveillance Center, Research and Development Division (NRaD). The Training Systems Division of the Naval Air Warfare Center (NAWC) has also contributed to the effort. The basic aim is to build a culture of teamwork and a unified framework for total ship system engineering (TSSE) that reflects a team approach.

This article is organized into four sections. The first section (Task and Drivers) lays out a framework for problem solving. The second (Traditional Ship Engineering Process) covers the existing way of doing business, while the third (TSSE Approach) deals with a proposed new approach. The fourth section (Vision and Opportunities) considers ways of moving toward implementation of the target framework.

Task and Drivers

Task

A sense that better teamwork could lead to better ships, as opposed to addressing a particular problem, brought "the gang" together. A number of serious concerns must be overcome to meet the emerging challenges of the next century. First, we face a period of austerity and change as U.S. forces are downsized. We also face considerable uncertainty due to the diversity of threats and environments expected in future littoral warfare operations. At the same time, it must be recognized that this is an era of rapid change in warfare systems and methods. Given the complexity of war-fighting systems, superior system engineering will be necessary to create new

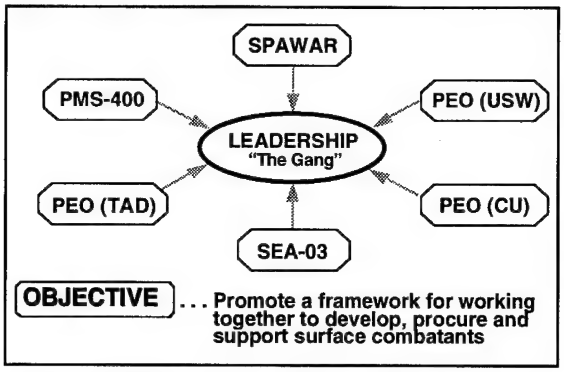


Figure 1. *Participants of effort, etc.*

combatants, tailored to the joint and expeditionary war-fighting concepts now beginning to emerge. Thus it is essential to reengineer the process by which operational requirements are transformed into combatants and war-fighting systems.

Drivers

As shown in Figure 2, major drivers for process design include: (1) ensuring that ship design is grounded in what the war fighters must do; (2) overcoming the problem of stovepiping in the development organization; (3) fostering use of modern system engineering methods to create ships that are both capable and affordable. A suitable framework is expected to consider agreed-to design concepts or target architectures, standards, and engineering tools. These would be used by development programs to ensure that their individual pieces are integrated into the total ship design and are applicable in other ship classes as well.

What Future Ships Must Do

We can think of ships in terms of the mission teams that they carry to the operating area, where they may be called upon to maintain presence, or to deliver high-tech firepower against an adversary. The most important outcome desired is to build ships that accomplish what the mission teams need (see Figure 3).

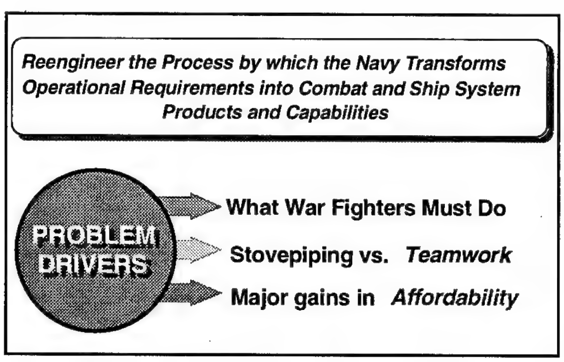


Figure 2. *Drivers for process design.*

To support mission teams in such environments, ships must be very flexible, capable of tailoring basic capabilities to the designated set of mission tasks and operating environments. In future conflicts, these teams will face multiwarfare threats in difficult (natural) environments, high levels of ambiguity, and complex rules of engagement. Participation as an integral part of joint and combined expeditionary forces will be emphasized. (This, in itself, means reinventing the process by which military requirements are transformed into ships, as stovepiping concerns increase with the level of force integration.) Due to the pace of technological progress, the ongoing "revolution in military affairs," and the importance of open systems for affordability, ease of upgrade must also be emphasized.

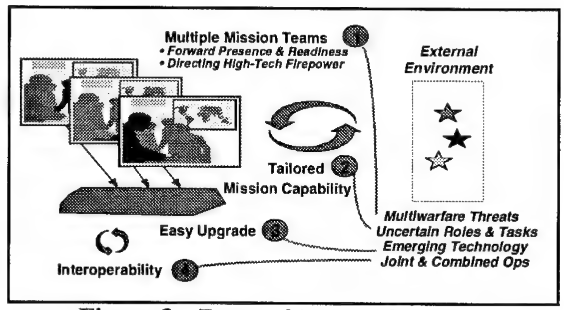


Figure 3. *Future ship considerations.*

What Developers Must Do

With respect to TSSE, the Navy is faced with a number of challenges. In our acquisition culture, systems (a) tend to mirror how we are organized (see Figure 4), (b) are developed independently with distinct elements, and (c) are procured as commodities for integration into larger systems. Given these characteristics, creation of well integrated systems demands a major effort. We have been able to make some progress by working hard to overcome the barriers—but have been only partly successful. The level of effort necessary in the current environment may not be affordable in the future.

The alternative is to build systems that work together, so that the ship as a whole becomes a well integrated "system of systems"

Pursue Development of a Total Ship Engineering Process That Is Applicable Across Ship Classes....

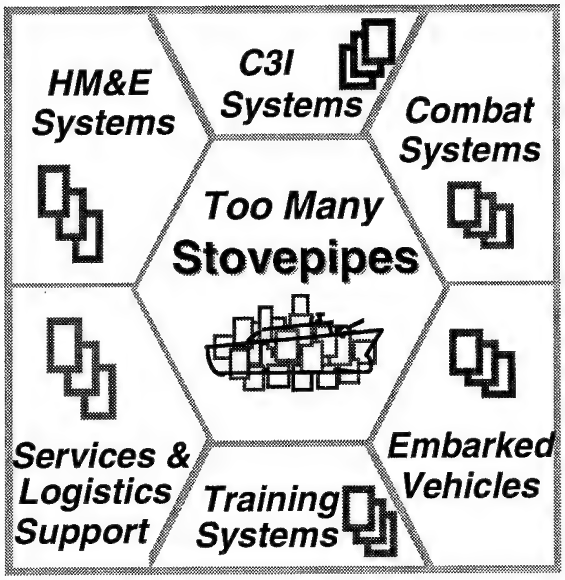


Figure 4. *Current ship system engineering represents stovepiping.*

(see Figure 5). This can be achieved by creating, between the ship level and that of individual systems, some kind of framework for coordinating development across individual systems. The top-level partitioning of control functions for the ship is seen as a key factor. The core problem is not how to break a ship into individual systems, but how the systems can be made to work together as parts of a unified whole. The fourth section outlines our approach to partitioning, which involves three elements: combat control, plant control, and information management. The first two involve some change in the traditional partitioning of combat systems from hull, mechanical, and electrical systems. The third element recognizes that information is a resource that demands coordination on a shipwide basis, and that special attention may be needed to create the necessary means for coordination.

Strategy for Affordability and Capability

A third major element of strategy has been to foster a disciplined system engineering approach in which affordability and capability are considered as two sides of the same coin (i.e., a strong fleet). The aim is to achieve major gains in affordability without sacrificing needed capability. This can be achieved only by reinventing the enterprise to improve productivity across the board. This is something the development organization must do.

Figure 6 applies to life-cycle cost for a typical ship. The relative size of the pieces will differ for any particular ship type. The largest piece here, personnel cost, can be addressed by automation and by process reengineering. While the original data applies to the ship's crew, it

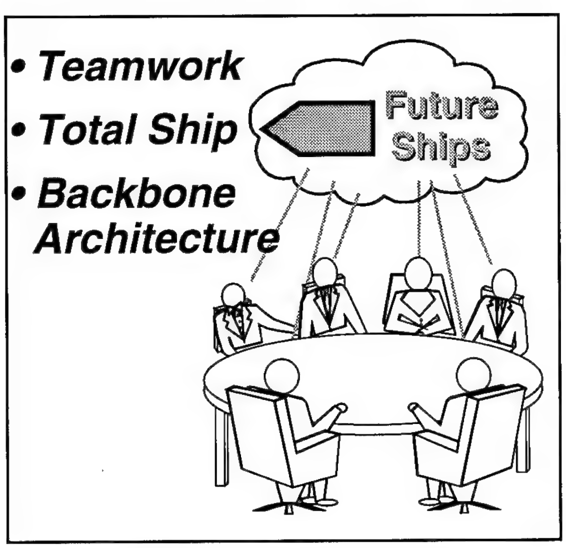


Figure 5. *System of systems.*

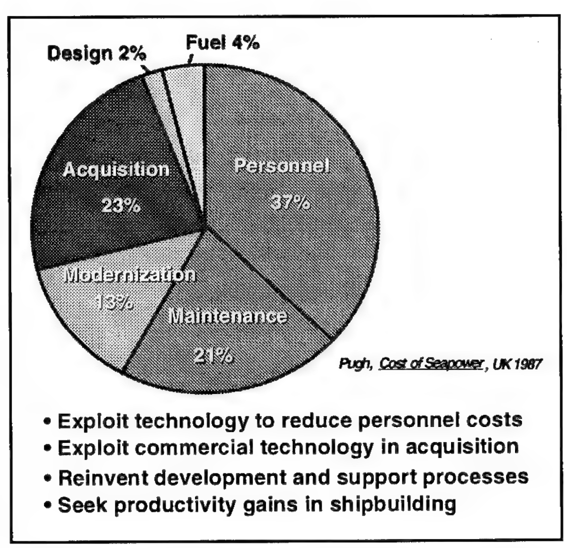


Figure 6. *Illustrative LCC Breakdown - naval ship.*

should be recognized that ashore personnel costs are also a major contributor. Other areas can be addressed by exploiting commercial products and by reinventing development and support processes. There is also potential for broad gains through improved productivity in shipbuilding and construction.

Current Efforts

A number of actions are being taken to pursue our goals. Reports have been drafted to lay foundations for a TSSE process (Figure 7) and to give results of a pilot effort in the combat control backbone area. A strawman set of shipwide design guidelines has also been circulated (in-house). The aim has not been to publish final results but to draw different parts of the surface ship community into a dialog.

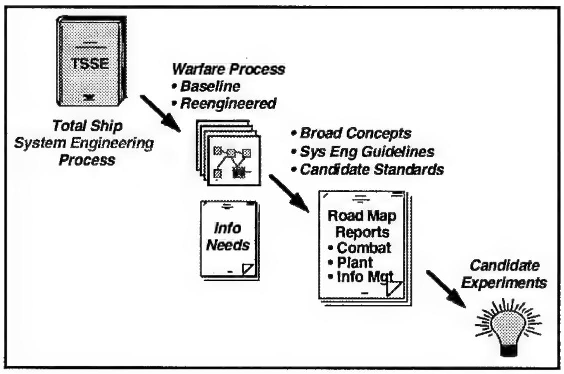


Figure 7. *In pursuit of goals.*

Efforts to extend the pilot phase to the remaining areas (plant control and readiness, information management) are in progress. Defining a backbone control structure for information management calls for an understanding of the information flows involved in key war-fighting processes. These will be considered in a reengineering workshop for selected mission teams and tasks. Within a short time, we should have a set of design concepts, standards, and tools for application to specific programs. However, interim results have been reported to a variety of headquarters activities. Periodic reports have also been given at American Society of Naval Engineers (ASNE) and Surface Navy Association meetings. Some of the material was given at the 21st Century Surface Combatant (SC 21) Industry Briefing in November 1994.

The next section of this article considers the existing process for total ship engineering. The overall structure of the process, its history, and the underlying concept of organization will be reviewed.

Traditional Ship Engineering Process

History

As Figure 8 suggests, the Navy has experimented with many different approaches to ship design. Stovepiping and the increasing complexity of modern systems have been perennial concerns throughout this period. In the 1950s, design functions were largely performed in-house. The Bureau of Ships (BuShips) had separate organizations for preliminary design and contract design. There was some in-house construction at naval shipyards. In the 1960s, the concept of total package procurement shifted design of several major ship classes to private shipbuilders. Use of public versus private sector shipyards for new construction was eliminated.

In the 1970s, there was a return to in-house ship design with a central design management organization in the Naval Ship Engineering Center (NAVSEC). However, the design to cost philosophy was practiced for much of the decade. The growing Soviet naval threat became a dominant theme in the mid-1970s, causing a great deal of attention to be paid to concepts and methods of ship design and engineering.

Acquisition streamlining became a major theme in the 1980s, with increased shipbuilder participation in the design process. This trend

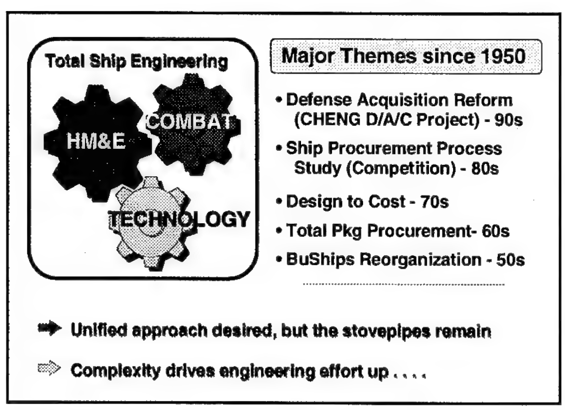


Figure 8. *Concerns and approaches to ship design.*

has continued into the 90s, with defense acquisition reform and commercialization emerging as major themes.

Baseline Process

The sequence of events shown in Figure 9 applies in a general way to naval ship design since 1970. For many years, the overall strategy for ship design, construction, and support has been dominated by a bottom-up approach to design and development. Ship designers choose the hull-form and propulsion machinery for desired maneuvering and sea-keeping qualities, while ship size and arrangement are varied to meet demands for space, weight, aperture, and stability driven by the required payload systems. The basic concept is to deal with problem complexity by dividing component development responsibilities among a loosely coordinated array of programs. Each program office builds a little, tests a little, specifies and, finally, produces a stand-alone system.

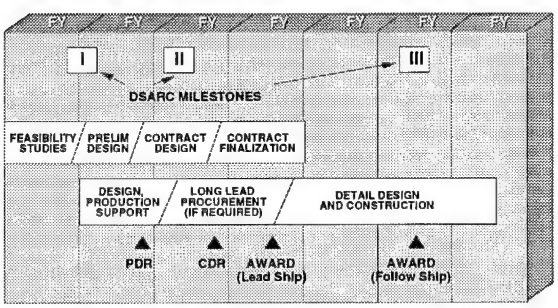


Figure 9. *Naval ship design since 1970.*

The ship is then constructed by a process of component assembly and integration, proceeding from the keel up. Ship designers have traditionally acted as physical integration agents, providing the hull, mechanical, and electrical interfaces necessary to package the stand-alone systems into the hull. In this context, total ship engineering means ensuring that appropriate ship elements are selected and effectively and efficiently combined to satisfy ship design requirements and constraints.

While this approach yields systems that work, it is costly in terms of acquisition, manning, and logistics support, and has made it difficult to achieve a highly integrated control structure for either the ship or the shipbuilding organization. However, today's computing capacity is virtually unlimited—freeing the designer to engineer the ship as a "system of systems." Qualities of firepower, stealth, interoperability, and affordability needed for future warfare environments make a top-down, integrated design process imperative.

Underlying Concept of Ship Development Process

Despite the many twists and turns in acquisition policy, the core process and main features shown in Figure 10 and Table 1, respectively, seem to have been fairly stable over time. To begin, the strategy is one of centralized planning and decentralized execution.

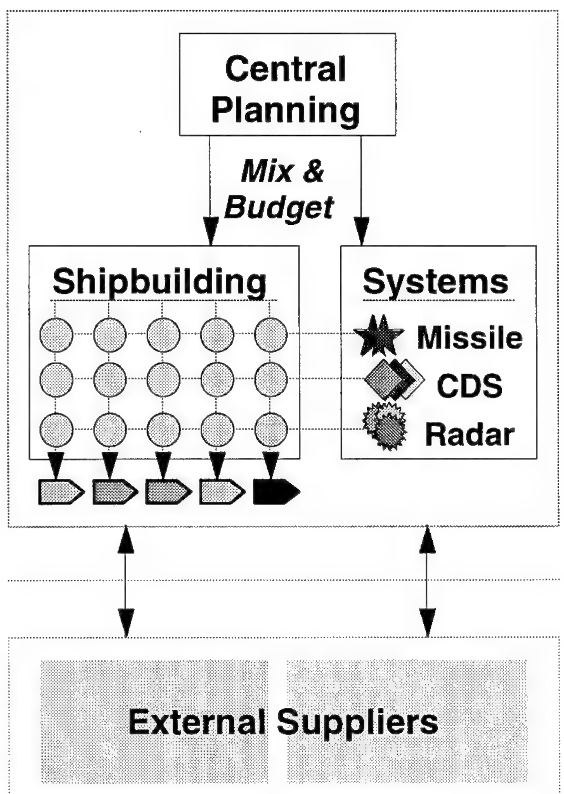


Figure 10. *Core process.*

Table 1. *Main Features*

- **Centralized Planning & Decentralized Execution**
- **In-House Team Leader**
  - *Control Design Process*
  - *Collocated Design Team*
  - *'Converging Programs'*
  - *Built-in Stovepipe Effects*
- **Many External Suppliers**
  - *Detail Design & Construction*
  - *Many Individual Systems*
  - *Integration After the Fact*
  - *Vulnerable to "buy-in"*

OPNAV plans the force, and NAVSEA executes programs. Ships are designed by a central engineering staff, with detail design and construction executed by a shipyard.

The core process resembles the "command and control" scheme pioneered by the General Motors Corporation to produce a car for "every purse and purpose." Many ship types are produced, each with different mission critical systems but sharing many common components as well. Where possible, common designs may be used across the entire fleet. A typical ship has scores of individual systems, each with a specific purpose, and thousands of functionally complete components. Scores of acquisition programs and thousands of suppliers are involved in creating components and delivering them to the shipyard. In the reference process, the central engineering staff designs the parts and gives the drawings to suppliers for bid. Some of the suppliers are in-house activities, and bureaucracy is a factor in dealing with them. Price is the main factor in dealing with external suppliers, and the process is vulnerable to buy-in.

The reference process is intended for in-house execution, and where used by industry, 70 percent of total value may be produced by in-house activities. The other 30 percent includes bulk materials and commodities (such as fasteners) that are widely available. In naval construction, teams of in-house activities and external suppliers are used for each system and ship, so the in-house value added is comparatively low.

The next section describes an alternate approach to total ship engineering that can be pursued to achieve desired outcomes.

TSSE Approach

"System-of-Systems" Framework

For a large composite system, such as a ship, dealing with component systems one by one is not good enough. A better approach is needed, one that permits coordination of many independent projects to create a fully integrated "system of systems." Use of a generic framework, dividing the effort into several projects and two levels of management, is suggested. Figure 11 shows such a framework. One level provides coordination for the overall system, working across projects, while the other manages individual projects. Establishment of this framework begins with domain analysis to establish a basis for specifying requirements. Next, infrastructure and integration architecture are used to enable integration of component systems into a consistent overall system in an efficient way.

System-of-systems engineering involves two kinds of integration, one that focuses on mechanisms used to interconnect parts and another that focuses on the uniformity of an overall system design. A dictionary definition of the term integration (to make whole or complete by bringing together parts) succinctly captures this tension between the parts and the whole. On the one hand, we talk about a system—a whole, or complete, thing; on the other, we talk about bringing together parts. The system side of the equation emphasizes global system properties, such as a harmonious "look and feel" in user interfaces, or the linguistic elegance of system structure. The parts side of the equation emphasizes interconnection and interoperation (e.g., of functions, data files, and subsystems).

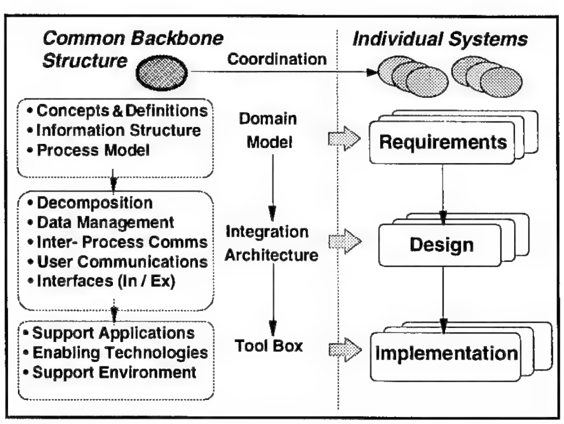


Figure 11. *Generic framework for "system of systems."*

Framework

As implied by the task statement, the missing step in today's approach to TSSE is a framework for dealing with the ship as a "system of systems" (see Figure 12). In other words, what is lacking is a way of defining and controlling the system engineering process across the entire set of independently designed and procured components necessary to build a surface combatant. Some elements this framework should strive for include: (1) a fully integrated control structure; (2) the ability to share system resources as necessary to coordinate and support subsystems; and (3) the ability to change and upgrade components more easily.

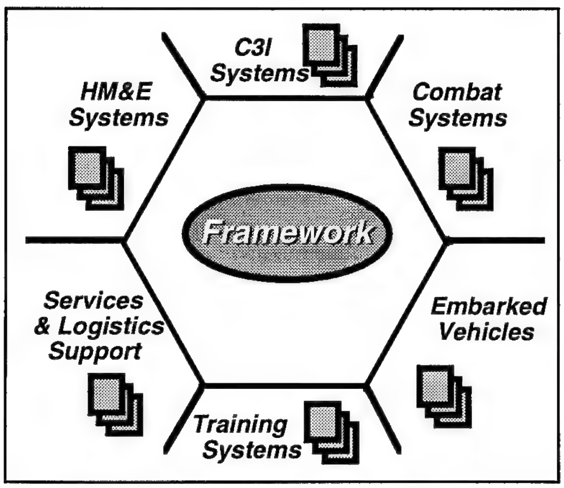


Figure 12. *Framework lacking in TSSE's approach.*

Point of Departure

Regardless of the approach to spatial arrangement and physical modularity, a control structure is necessary to make the ship responsive to command direction and control. In defining a framework for dealing with command and control functions on a total ship basis, we start with a simple view of the ship as shown in Figure 13. People are shown at the top, organized into mission teams. Only by the direction of its crew does the ship become a complete war-fighting system, capable of acting on its own to some degree. Even a fully automated ship would execute broad plans and orders only in accordance with direction from a mission team located elsewhere. The mission teams are supported by various categories of physical resources (radars, ship machinery, missiles, etc.), accessed via

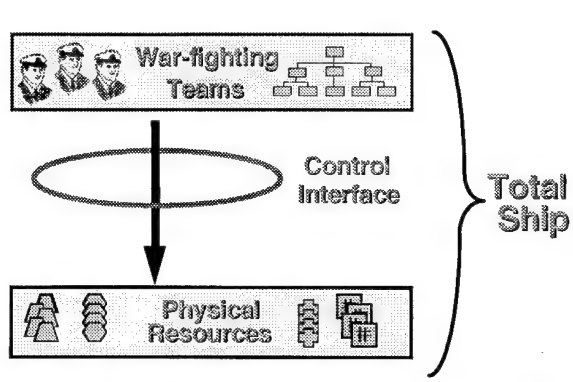


Figure 13. *Simple view of ship control structure.*

control interfaces. What is being addressed in the system-of-systems framework is the process by which the control interfaces are engineered. An expanded view appears in Figure 14.

The strategy for partitioning control resources on a total ship basis is rooted in basic principles of ship and combat system engineering. The first principle is that the aim of design must be to help the war fighters in achieving their operational objectives. The implication is that a partitioning for total ship design must be driven first and foremost by operational considerations. Another important principle is that engineering work units should be organized on the partitioning thus determined. The control structure thus reflects two key viewpoints. The first must be a war fighter's view, considering the essential military purpose of the ship. The second is that of the development organization responsible for ship design, acquisition, construction, and support.

Total Ship "System of Systems"

The operational viewpoint is considered first. The center section of Figure 13 is here expanded

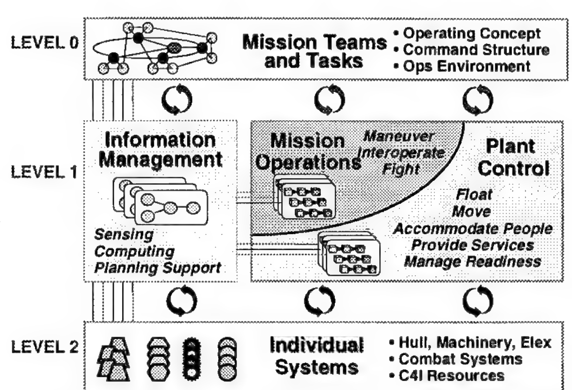


Figure 14. *Expanded view of ship showing structure of control interface functions.*

a bit to show the basic structure of the control interface functions. How the control structure is partitioned, and how the parts interact, is the key to total ship system integration.

Combat control (mission operations) and plant control involve a rearrangement of the traditional partitioning of combat systems from hull, mechanical, and electrical systems. However, in future ships the position of this boundary may change. Maneuver control and damage control coordination are seen as increasingly important concerns that may cross the boundary as indicated. The third element of this partitioning—information management—recognizes that information is a resource that demands coordination on a shipwide basis, and that special attention may be necessary to create suitable means of coordination.

The three major subsystems we refer to as "control backbones." Some of the major functions to be controlled in each are shown. As a system of systems, the ship provides mission teams in each area with the resources to perform all assigned tasks, together with the interconnections and interfaces that make them responsive to human direction and control. For ships that are not combatants, the term *mission operations* can be used in place of *combat control.* With this convention, the partitioning shown above is applicable to all ship types.

Sequence of Design Decisions

The core of the systems engineering problem is not how to decompose the overall system into subsystems, but how to integrate subsystems into an overall solution. The challenge is to decompose tasks and to allocate subtasks without compromising the wholeness of the task. When the problem is viewed from this perspective, it is clear that a system engineering process involves a set of interacting (or interdependent) subproblems. The sequence for addressing the subproblems, together with the solution strategies employed, becomes a specification for the process.

Figures 15 and 16 represent two different descriptions of a common process. Although problem solving may follow a spiraling and iterative trajectory through the different sets of subproblems, the solution at any stage cannot be finalized until solutions are reasonably well worked out for previous stages.

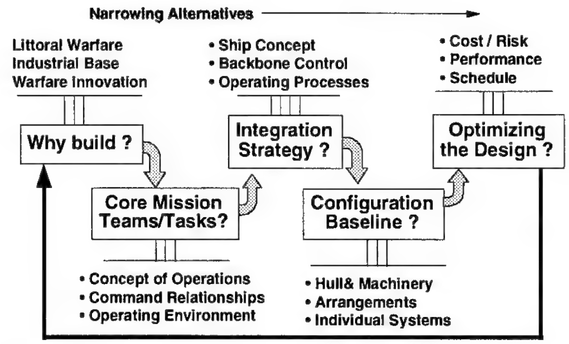


Figure 15. *Linear concept for Total Ship System Development.*

TSSE

The development organization must be structured to maximize value delivered to mission teams on a life-cycle basis. This drives the role of the development team leader, which must be concerned with the enterprise as a whole, working to shape the value stream to maximize the value delivered.

Activities shown in Figure 16 are seen as core concerns of the development team leader. Because it must deal with the acquisition and integration of individual systems as well as overall ship construction, team structure is a bit more complicated than the operational control structure. Once an adequate understanding of mission teams and tasks has been created, the development team must address ship design, the design of backbone control structures, and the definition of specific operating processes, adequate to meet mission needs. Each makes a key contribution to creation of a total ship "system of systems" from the variety of individual systems delivered to the shipyard.

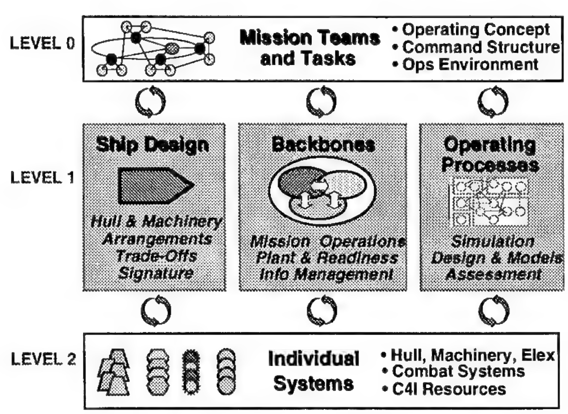


Figure 16. *Layered concept for Total Ship System Development.*

The ship design activity provides the engineering studies necessary to address the questions of ship form, size and essential military characteristics. Speed, sea-keeping, strength, stability, and style are the major naval architecture issues. These characteristics are all interrelated and dependent on overall ship configuration and dimensions. Accordingly, this activity controls the ship design budgeting process, including signature and survivability in addition to weight, space, moment, and cost factors. It also provides the physical interfaces (spatial, mechanical, and electrical) necessary for the many individual systems packaged into a multimission ship. Virtually all the resulting information finds its way into the drawings and specifications used by the shipbuilder.

The backbone design activity provides for integration of all onboard control resources, in effect making the ship a real "system of systems." Since ships are composed of many individual systems, the control structure must include mechanisms to facilitate cooperation among them, without compromising their ability to perform mission tasks. Just as domain models permit a common understanding of goals and requirements at the level of individual systems, the backbones permit a shared understanding of how control functions and interfaces will be handled. Their creation will establish a standard for services available to individual systems, and provide guidelines for the design of control structures and interfaces by individual projects.

An activity responsible for process evaluation and integration is also necessary. The basic purpose of this activity is to understand what mission teams must do and how well the ship, as a system of systems, can create and coordinate action paths adequate to meet their needs. This calls for a total ship perspective, together with a capability for detailed analysis of ship characteristics and performance.

Seeking an Integrated Value Stream

The effort to reinvent surface combatants doesn't stop with the TSSE process. The Navy can also reinvent how it organizes for design, construction, and support of future surface combatants. We do not have a specific solution to recommend, but have begun to consider this question.

In fact, industry has turned reinventing the enterprise into something of a global trend. For the most successful efforts, thinking in terms of a value stream has been the crucial first step. The enterprise is defined not as a single activity but a group of activities working together to supply goods or services in a way that creates maximum value for the customer (in our case the war fighter or mission teams). This drives the role of the enterprise leader, who must act to shape the overall value stream and not give value added by direct individual effort alone. This causes a shift away from stovepipe thinking to global or team thinking. In particular, greater attention is given to relationships among team members and to the transactions between them that often have the most potential for effectiveness and productivity gains. Figure 17 indicates how such an enterprise might be structured.

This approach is widely viewed as an improved model for designing large productive systems. The original implementation is credited to Eiji Toyota and Taiichi Ohno and is sometimes called the Toyota Production System. The basic aim was to form a vast group of suppliers and parts plants into one "machine" by producing at each step only those parts necessary to satisfy immediate demand at the next step. The final assembly organization functions as enterprise leader. Usually, design and production of components that tend to define product style and performance (the product's "signature") are

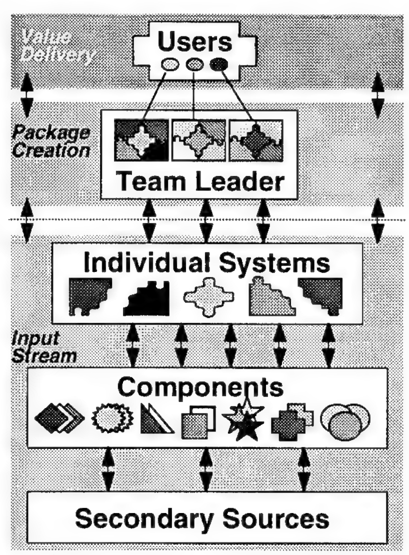


Figure 17. *Enterprise team thinking is key to value stream.*

supplied in-house. Thus the enterprise maintains control of the product line, but the value added by in-house divisions may be as little as 25 percent of the total.

Suppliers are organized into functional tiers, with multiple products and multiple sources in each tier. First-tier suppliers play an integral role in product development and are assigned a whole component to design. The suppliers work to a performance specification for a system that must work in harmony with other components from other suppliers. Toyota formed first-tier companies by spinning off in-house divisions and by building long-term alliances with external suppliers. Production is typically shared among several sources, with shares fluctuating up or down according to performance.

Second-tier suppliers tend to specialize in a manufacturing process. A first-tier supplier might design an alternator, for example, and buy all parts from second-tier suppliers. The latter have no role in overall product design but may produce drawings for individual parts and have firms at still lower tiers produce parts to those drawings. Since companies at level 2 generally do not compete for specific component types, they can work together in supplier associations for the purpose of sharing information on manufacturing techniques.

The concept of operation is based on mutually agreed upon pricing, strong incentives for performance and sharing of information, and long-term relationships. Direct competition for production work between in-house and external activities is avoided, as it tends to be inefficient and unfair.

For surface ships, maximizing value delivered to mission teams on a life-cycle basis would become the basis for organization. The enterprise leader is viewed as an Integrated Product Team with both Navy and builder elements (see Table 2). The activities shown in Figure 16 are seen as core concerns of the lead activity. In short, the enterprise leader controls the overall design process, including weight, space, and cost budgets; the strategy for integration and control of mission capability; and creation of the mission-critical systems that are the reason for taking the ship to sea.

Beyond this, we believe the Navy should rethink the entire chain of supply (see Table 2),

Table 2. *Key Questions*

⇨ Establishing User Needs
- *What Must War Fighters Do?*
- *Role of Surface Ships*

➡ Role of Team Leader:
- *Vision of Total Value Stream*
- *Design & Budgeting Process*
- *Integration Strategy*
- *Mission-Critical Systems*

➡ Rethink Chain of Supply
- *Multiple Streams & Sources*
- *Performance Incentives*
- *Open Specs & Standards*
- *Long-Term Relationships*

adopting the best practices from major enterprises around the world. At each tier, supplier associations can be formed to create open specifications and standards; process improvement techniques can also be shared. Suppliers in the first tier (major systems) should participate in overall ship design. Lower-tier suppliers should be able to participate in both commercial and defense markets. Ideally, Navy research and development (R&D) results would be shared as much as possible among same-tier activities.

Teamwork Environment

Teamwork is one of the key characteristics sought in the target TSSE process. This calls for use of a system-of-systems engineering approach and reliance on shared concepts, standards, and tools to promote design integration. Probably the first and most important step is to form a cadre able to consider tradeoffs from a total ship perspective. At the same time, full appreciation for (and access to) the specialized technical knowledge of functional activities remains essential. Teamwork is very difficult because the problem is very complex. By fostering a shared understanding of the problem and adopting a common language, cadres can learn to communicate well enough to permit good teamwork. Once formed, cadre members can work effectively together from different locations, as long as frequent communication is permitted.

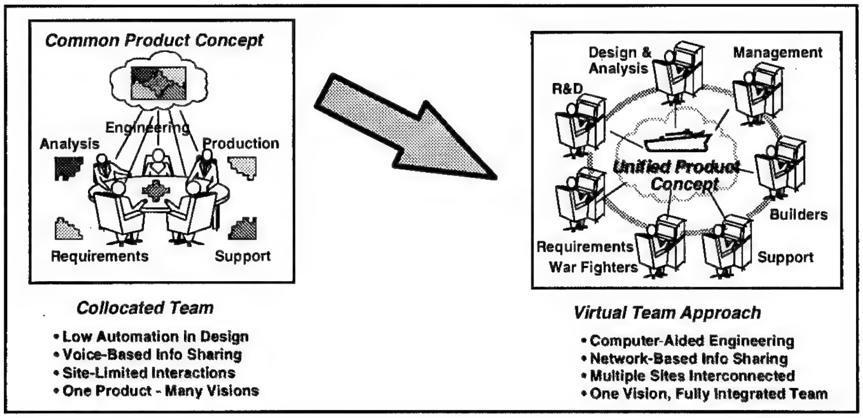


Figure 18. *Transition from collocated to virtual team approach.*

The idea of using computer networks for teamwork is no longer a new one. Today it is possible to think of a team being brought together on the network, sharing information, and holding meetings entirely via the network using common design aids and groupware. A Virtual Team (see Figure 18) is an Integrated Product Team consisting of members with different perspectives of the product development, wielding powerful computer-aided design (CAD) tools in their individual workspaces but publishing results to a shared information base. An overarching coordination service may also exist to schedule work, report progress, notify persons, generate work authorizations, and carry out entire suites of coordinated processes without the need for face-to-face meetings.

The next section considers what opportunities exist currently for pursuing the broad actions identified in this section. It is essential to recognize that ship development tends to be evolutionary in character. Ships are so complex that it is difficult to create entirely new combatants, with no use of legacy systems.

Vision and Opportunities

Begin With Mission Teams

As indicated earlier, mission teams and tasks are the starting point for TSSE. Within this context, efforts are now under way to consider methods for determining what future surface combatants must do, what corresponding warfare processes will be like, and what information will be needed to execute those processes. The basic concept is to form several independent warfare process teams (see Figure 19). Each team will consider a single operating domain and produce two main deliverables: (1) a vision of the future warfighting process and associated reengineering opportunities; and (2) a summary of information flows necessary to implement the vision process. Tentatively, warfare teams will be set up to address expeditionary warfare, maritime fire-base operations, joint air dominance, and integrated survivability. These represent only a subset of the mission teams that must be considered. However, it is a subset that captures many of the changes that appear needed in future combatants. In expeditionary warfare, what Amphibious Ready Groups must do is the topic of interest. In this case, the subject mission team is not a single-ship team but a force-level entity. A fifth team will deal with core concerns such as team decision-making, enhanced operator interfaces, common (joint) data structures, and open systems.

A key goal for total ship engineering process improvement is to strengthen the sense of partnership between mission teams and the development and support teams that stand behind them. Primary emphasis here is on understanding what the mission teams must do

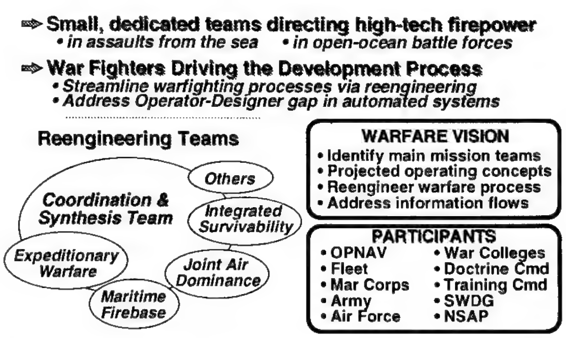


Figure 19. *Warfare process teams to reengineer vision and mission.*

and engineering systems to support or execute those tasks efficiently. Future ships will fight by wire! The control backbones, which consist of displays, computers, application programs, etc., interface the war fighter with the resources, weapons, sensors, etc., that carry out the mission. The backbones determine not only the "look and feel" of the ship to the war fighter, but will have embedded in them how the war fighter will fight the ship in terms of processes and procedures. It is, thus, imperative to have the war fighter involved in the design process. The designers must listen to the "voice of the war fighter," the customer.

Reengineering Example

An example of process reengineering in the private sector helps to show the potential value of the approach. In the early 1980s, an American automaker put its accounts payable function under the microscope. This department had over 500 people at the time, but management believed that with new computing systems, the same job could be done with only 400 people. This sounded good until it was learned that a similar company needed only 5 people for the same job. Soon it was recognized that much better results could be achieved, but only by rethinking the entire process of procurement rather than the accounts payable function alone. The procurement process, as diagrammed in Figure 20, took as input a purchase order from a plant needing parts and provided the plant with bought-and-paid-for goods. It involved not only accounts payable but also purchasing and receiving tasks.

Process execution began with the purchasing department sending a purchase order to a vendor, with a copy going to accounts payable. When the vendor's shipment arrived, a clerk at the receiving dock completed a form describing the goods and sent it to accounts payable. The vendor's invoice also went to the department which, thus, had three documents (receipt, purchase order, and invoice) on the transaction. Payment was quick if they matched, which was most of the time. But most of the department's work went into the occasional mismatches. Some cases took weeks and enormous amounts of work to resolve.

The new process (shown in Figure 21) seeks to eliminate mismatches altogether. When the purchasing department issues an order, the information is entered into an on-line database. Vendors send goods to the receiving dock as before. When they arrive, a receiving clerk checks the database to see if they correspond to an outstanding purchase order. If so, the clerk accepts shipment and records the transaction. Under the old procedures, 14 data items had to be matched among the receipt, the purchase order, and the invoice before payment could be made. Now only 3 items are checked, and the matching is automated. Vendors no longer send invoices. When the goods do not match an outstanding purchase order, the clerk on the dock will refuse the shipment. Results are dramatic—in fact, the new process comes close to eliminating the need for an accounts payable department altogether.

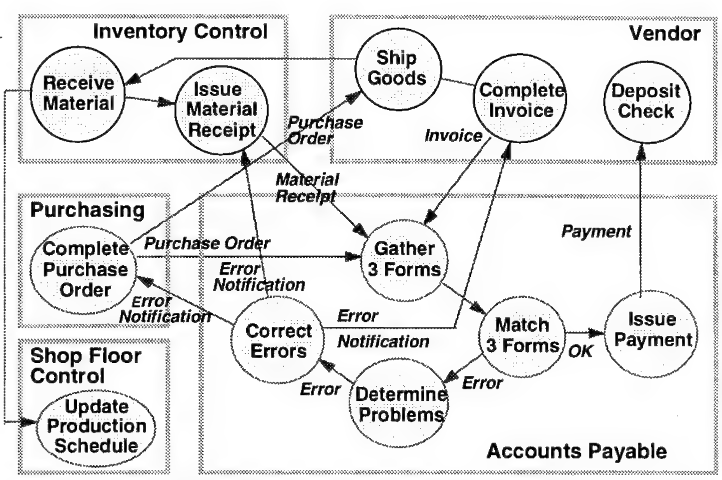


Figure 20. *Business process - baseline.*

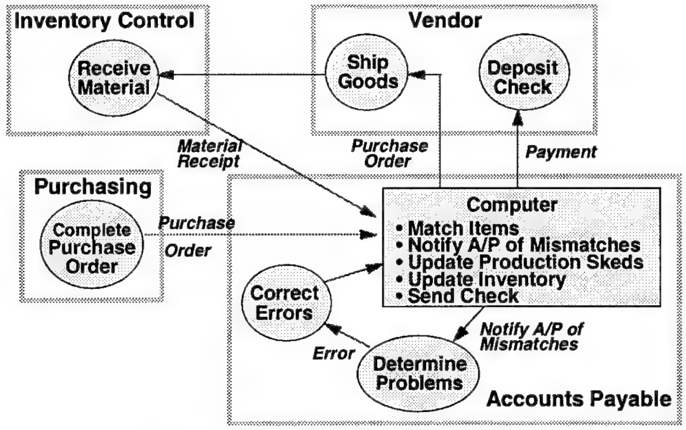


Figure 21. *Reengineered process.*

Reengineering Mission Teams

Questioning the need for three different descriptions of each transaction was the key step in the above example. The same question can be asked with respect to war-fighting processes. Figure 22 shows the primary external interfaces and key internal processes for a ship acting as a maritime fire base: i.e., delivering ordnance against land targets of all types. The ship will have to deal with various cooperating commands and units, each of which may supply a piece of the overall tactical picture. The players and relationships are likely to change as new doctrine and capabilities for joint and coalition warfare emerge. In addition, the ship deals with different target sets in amphibious fire support, strike, and antisurface operations. Since the target sets overlap, it is not immediately clear if there should be three different (single-purpose) target databases or one multipurpose database. In fact, target data from forces ashore is yet another set that might be integrated with the existing three databases. The aim of reengineering is to go back to the beginning and invent a better way of conducting such operations. This may involve better ways of responding to change in joint command structures and better ways of handling target data.

Similar opportunities exist in other broad mission areas. In joint air dominance operations, several different track databases may exist, reflecting differences in source and target type (see Figure 23). Ballistic and cruise missiles, fixed wing and rotary aircraft, and unmanned air vehicles are target categories with different tracking, identification, and engagement characteristics; hence it is not immediately clear how best to organize the track data. In addition, each major weapon system tends to produce its own approach to tracking. Efforts to reengineer Joint

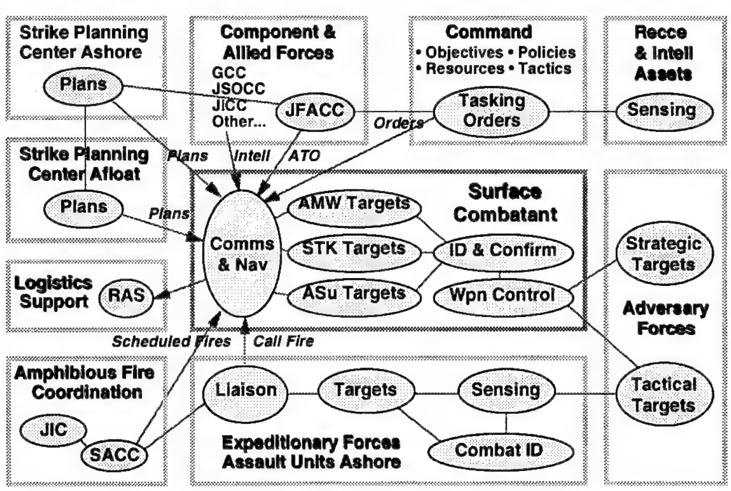


Figure 22. *Maritime fire base team.*

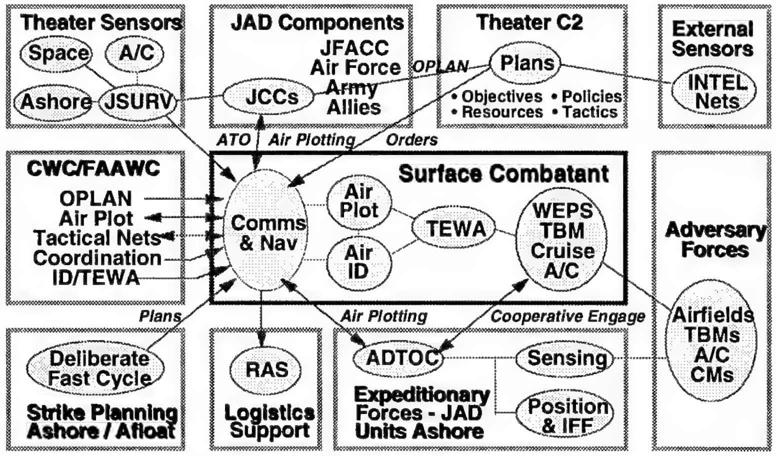


Figure 23. *Joint air dominance team.*

air dominance processes would reinvent associated control and information flows. Survivability is another area of interest, because it calls for a balanced view of both own ship and threat status to maintain situational awareness from a total ship perspective (see Figure 24). In short, reengineering has potential for almost any mission team and offers a useful strategy for rationalizing mission information flows.

Vision for Command Spaces

Littoral warfare operations seem likely to demand increased flexibility in surface ship command and control capabilities. For example, a future combatant might be organized to deal with power projection, battle space dominance, information warfare, survivability, and mobility as the major operational tasks to be performed. In addition, various special purpose teams may have to be supported. This might be a joint air identification team; or it might mean integrating a U.S. Marine Corps (USMC) air defense control team (using embarked rather than organic facilities) into the ship's combat control structure. It could also be necessary to adapt to changes in the joint command structure as a conflict situation evolves. Indeed, each forward deployment cycle might call for a different command structure or variation from some core structure.

A design that makes physical rearrangement of mission teams and watch stations easy is then important (see Figure 25). The initial design should be viewed as the nucleus of a more advanced or larger system, with hooks installed to support change. System engineering methods must be formulated to identify designs with a substantial measure of operational flexibility and to provide decision aids for tailoring a ship's command structure to specific mission needs. In time, major changes are likely in how military forces organize to use information. This could mean collaborative work styles, based on team workstations. Instead of hierarchies, in which

Figure 24. *Integrated survivability team.*

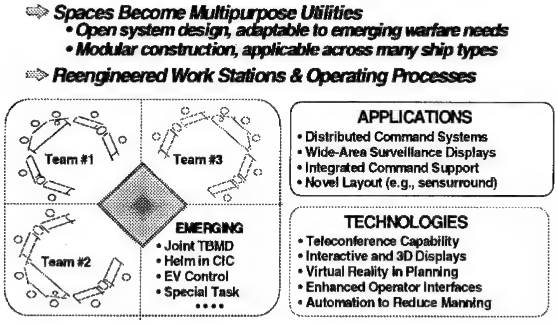


Figure 25. *Vision for command spaces.*

bottlenecking limits flexibility, future systems may use network structures extending across the lifelines. A third key source of change is automation, which can alter task allocation between humans and machines. To prepare the way for change, it is important to establish a disciplined process for managing life-cycle cost and system integrity.

Technology Opportunities: Displays and Portable Devices

Future watch stations will provide better displays, expanded feedback, and better ways to make decisions. Opportunities include the following (see Figure 26):

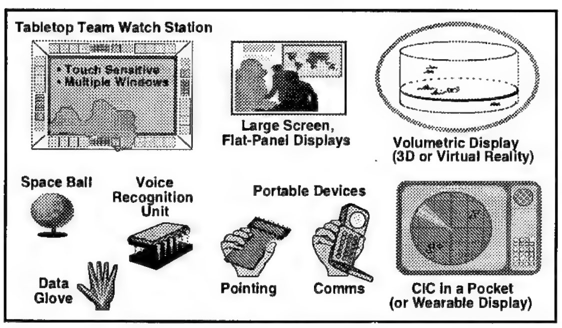


Figure 26. *Examples of innovations in human-computer interfaces, graphics, and enhanced communications.*

• **Human-Computer Interfaces.** How operators interact with displays can be much improved over today's hook/ball tab and cursor technique. Interactive device technology will permit the use of voice recognition, interactive screens, and a variety of pointing techniques in future display interfaces. Pointing techniques will include eye-safe laser devices, photoelectric light sensitive pointing devices, mouse-type devices, and others. There are existing prototypes that allow interaction with various displays via eye contact. This has been done to let pilots get information without using any other device. Interactive (touch-sensitive) screens exist today, and improvements in granularity and sensitivity will continue.

• **Graphics.** Future displays will use new techniques to aid tactical operators. This may include extensive use of color, windows, and icons; and new display heads. The latter may include large wall screens, tabletop units, or 3D displays. Volumetric (3D) displays will enable better use of the cognitive skills underlying human vision to better and more quickly understand the tactical situation. They will help to increase the amount of information that an operator can interpret and act on quickly and confidently. New symbol sets will permit use of icons in tactical displays, leading to reduced training time and better skills retention.

• **Enhanced Communications.** This will include the use of handheld, wireless intraship comms supported by an automated locator service.

Common Backbone Vision for Combat Systems

Based on the TSSE process and partitioning scheme outlined in preceding sections, reengineering opportunities have been addressed for the combat control area. The main question considered in this effort was whether it makes sense to talk about a common system engineering framework across many different projects in the combat system category.

Results indicate a common backbone structure may be feasible in future combat systems (see Figure 27). This would mean, for the combat system as a whole, the kind of flexibility and resource sharing achieved by the Vertical Launch System (VLS) in handling multiple missile types, or by the AEGIS Weapon System (AWS) in handling multiple simultaneous targets. The idea of a Common Operating Environment (COE) is usually applied only to a computing environment. Creating a common backbone for combat control means a COE defined more broadly, to include not only computer resources but also watch stations, interfaces, communications, displays, and mission support applications. Common system

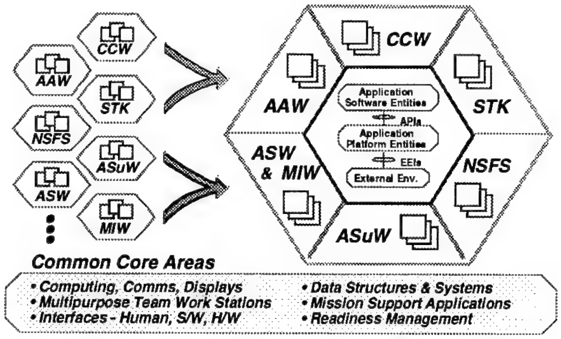


Figure 27. *Common backbone–vision for combat systems.*

services will be provided to deal with configuration management functions (e.g., naming and maintaining a common data element dictionary). The backbone would provide a standards-based framework for development and integration of individual combat systems. Individual system programs would adopt a narrower focus on delivery of mission-unique applications and components. Use of an open system framework can make this both affordable and capable. The potential benefits appear very significant.

Vision for Ship Integration

As indicated in the previous figure, the advantages of a common backbone are not limited to combat control, but apply to the area of plant control and readiness and the area of information management as well. What is envisioned is a generic backbone (with variants for each area) that supports design for modularity, commonality, and the sharing of functional resources on a shipwide basis. The generic backbone would provide the following characteristics:

- Command spaces become utilities, tailorable to any set of mission teams and tasks that may be operationally required.
- Computing, communication, and display resources are managed on a shipwide basis, with a common application environment maintained.
- Readiness and resources are managed on a shipwide basis.
- Life-cycle costs are reduced through efforts to hold manning and parts count to minimum levels, and by adopting an open systems approach.

The capabilities sought in the individual areas are identified in the bottom part of Figure 28. Overall, the resulting architecture is intended to be enduring and flexible to permit: (a) application to a variety of ship types and designs; (b) insertion of new functionality as war-fighting systems evolve; and (c) insertion of new technology as it becomes available. The original architecture should include an extension framework and be subject to formal change control from the earliest stages of development.

Total Ship Target Architecture — A Common Operating Environment

Figure 29 represents the ship as a layered open system with three entity types and two interface types. The target architecture represents a strategy for applying the concept of open systems to warship development. The qualities of portability and interoperability offered by open systems are combined with the reliability and effectiveness needed in combatants.

The target architecture is layered to form two loosely coupled subsystems. The first links application software entities to application platform entities. As in the Application Portability Profile defined by National Institute of Standards and Technology (NIST), the basic idea is to make the services provided by the application plat-

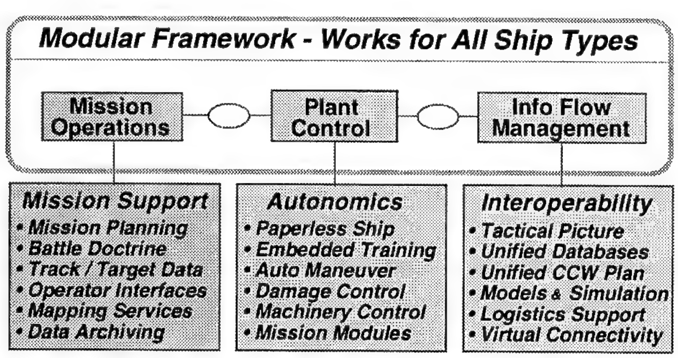


Figure 28. *Vision for ship integration.*

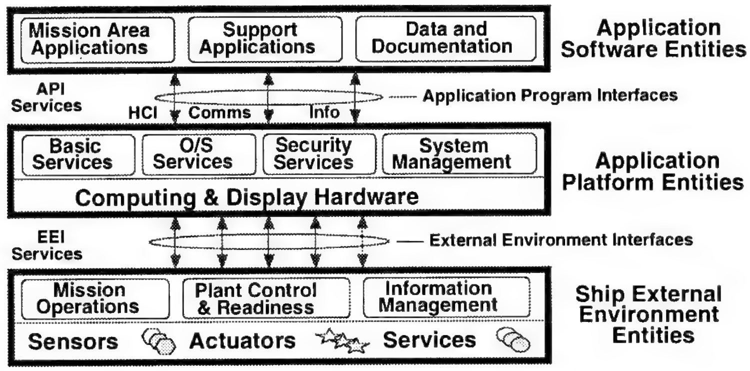


Figure 29. *Total ship target architecture–common operating enviroment.*

forms (at their interfaces) transparent to application software. This first subsystem is then loosely coupled in that application platform entities become interchangeable and application software entities become reusable.

The second subsystem links external environment entities to application platform entities. Again, the basic idea is to make services provided by the latter (at their interfaces) transparent to individual sensors, weapons, workstations, machinery, and service systems throughout the ship. Both application platform entities and external environment entities are viewed as providers of standard services, and any two elements providing equivalent services should be interchangeable.

An Evolutionary Strategy

The idea of a common backbone system is promising, but much remains to be done. One of the difficult areas concerns the lack of a working baseline with open-system characteristics. While reliable and effective, combat systems today are hardly open, and it is not yet clear how to deal with the problems of weapon safety certification for open systems.

Migrating to an open system involves an extra measure of risk that is important for both program executives and suppliers. Creating incentives for taking the associated risks is, therefore, an important problem. A related problem is to find ways to effectively manage standards-based backbone architectures, which can and must evolve with time.

Overarching Design Objectives
- **Working Baseline for a Shipwide COE**
- **Incentives for Migration to Open Systems**
- **Manage to a Standards-Based Framework**

Given today's budget constraints, a transition to common backbones probably won't happen all at once. However, the existing LPD-17 program offers a starting point. Progress made toward a common backbone structure in this program could be the foundation for full implementation in subsequent ship design and construction programs (see Figure 30).

Summary of Opportunities

The main characteristics sought in a TSSE process can be listed as follows:

- Process driven by what the war fighters must do
  - continuous dialog on mission tasks and system characteristics
  - ability to tailor configuration for designated roles and op areas
- Common backbones and building blocks
  - same control backbones applicable to all ship types
  - command spaces become utilities, useful for any mission task
  - plug and play flexibility of mission systems, data, and resource flows

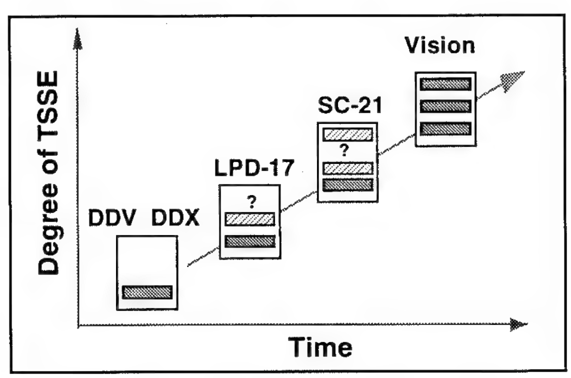


Figure 30. *Backbone migration strategy.*

- Open systems with extensive use of commercial off-the-shelf (COTS)
  - portable, scalable, reconfigurable, interoperable, extensible
  - easy upgrades for better performance, reliability, and flexibility
- Exploit potential for improved design methods
  - simulation-based design capabilities
  - reengineered mission teams and operating processes

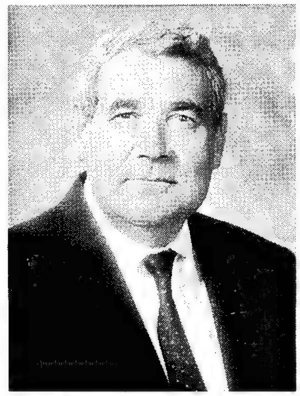
These characteristics will mean significant gains in Affordability

A good deal of attention was given to selecting the proper starting point. In the final analysis, we believe that the purpose of ships is to carry mission teams to a chosen operating area (at sea). What ships must do depends on the designated mission teams and tasks. System engineers must work constantly with the war fighters to define the necessary mission teams and tasks, and to engineer the operating processes necessary to carry out those tasks.

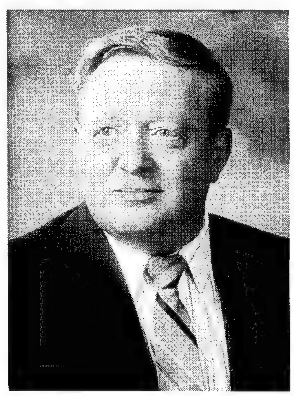
The advantages of a common backbone apply not only to combat control, but to plant control and readiness, and to the area of information management as well. What is envisioned is a generic backbone (with variants for each area) applicable across ship types. Use of common building blocks and open systems would be emphasized on a shipwide basis for all categories of systems (e.g., pumps, electrical systems) and not just control structure. Ships would thus have a minimum set of piece parts.

Finally, new engineering methods and tools offer great promise for improving the product (warships) and the process of warship design, acquisition, and construction. Opportunities in this area are especially important because, in a sense, warship design never starts with a blank sheet of paper—many components used in construction are built to earlier designs, and modernization during the life cycle may introduce yet later designs. Engineering principles and methods embedded in design aids will influence compatibility of designs from different decades.

The Authors



JAMES R. POLLARD, Principal Systems Engineer in the Dahlgren Division's Combat Systems Department, received a B.S. degree in physics from Roanoke College; M.S. in electrical engineering, George Washington University; and Ph.D in systems engineering, University of Virginia. After joining the Center (now the Dahlgren Division) in 1962, he performed research in antennas and electromagnetic coupling. He then managed the Special Effects Weapon Research Program, which focused on high-power electromagnetic devices. After serving as head of the Weapon Systems Division and, subsequently, the Search and Track Division, he headed the Strategic Planning Group. In 1985 he received the Bernard Smith Award for creative development and implementation of a Center strategic planning process. From 1985 to 1987, he served as Warfare Systems Architect in the Space and Naval Warfare Systems Command. In 1993 he received the John Adolphus Dahlgren Award for his work in total ship system engineering. Currently Dr. Pollard is conducting studies in future shipboard combat systems design.



BERNARD G. DUREN is an Operations Research Analyst in the Combat Systems Department of the Naval Surface Warfare Center's Dahlgren Division. He received an A.B. degree in mathematics from Indiana University and an M.S. degree in operations research from the Georgia Institute of Technology. His major duties involve the development of system concepts and system engineering methodology in the primary work areas of combat system architecture and system engineering methods. Prior areas of achievement include the startup of a strategic planning process at the Dahlgren Division, analysis of seaskimmer defense issues, and formulation of concepts for the defeat of antiradiation missile threats to Navy task forces. His academic interests include information integration, group problem solving, and strategic planning.

The Promise of Quantum Computing

Allen D. Parks

A decade ago, D. Deutsch established the foundations for the theory of quantum computation. This theory is necessary to study whether computing machines capable of harnessing quantum mechanical effects can solve problems more efficiently than classical machines. Although the theory is not yet completely developed, recent research strongly suggests that if quantum computing devices could be built, they could provide computational power that would far exceed that achievable by contemporary computing machines.

Introduction

The computations performed by modern, general purpose computers mimic those of the Universal Turing Machine (UTM). The UTM (see Figure 1) is a theoretical model of the most general idealized classical computer:

1. It has an unlimited memory.
2. It can perform with perfect reliability an arbitrary number of computational steps.
3. It is describable within the framework of classical physics.

However, since the universe is quantum physical, it is this last feature that suggests that the Turing Machine (TM) is an inadequate model for all physically realizable computing devices.

It is therefore natural to inquire into the computational properties of new types of computing machines based upon quantum physics. Early work in the field of quantum mechanics and computation was done by Benioff[1-4] and Peres.[5] Although they did not consider whether quantum phenomena could be harnessed to enhance computational power, they did show that quantum mechanical Hamiltonian systems could simulate the computations of a TM. Feynman[6,7] was the first to suggest that quantum computing devices might potentially be more powerful than TMs by observing that there is a possibly inherent exponential slowdown when simulating a general quantum physical system on a TM that might be avoided if, instead, a computing machine employing quantum mechanical principles were used. Deutsch[8] was the first to propose a model for the universal quantum computer and thereby establish the foundation for the theory of quantum computation.

The Universal Quantum Computer

In order to better understand the notion of quantum computation, we survey Deutsch's model. The components of a Deutsch quantum computer (DM) abstractly resemble those of a TM. A DM state is a normalized vector in the Hilbert space $\mathcal{H}$ spanned by eigenvectors $|x;\bar{n};\bar{m}\rangle = |x;n_0,n_1,\ldots,n_{M-1};\ldots,m_{-1},m_0,m_1,\ldots\rangle$ of the observables $\hat{x}$, $\hat{n} = \{\hat{n}_i{:}i\in Z_M\}$ and $\hat{m} = \{\hat{m}_i{:}\in Z\}$, where $n_i,m_i\in\{0,1\}$; $Z_M = \{0,1,\ldots,M\text{-}1\}$, and Z is the set of integers. Here $|\bar{n}\rangle$ encodes the internal state, $|\bar{m}\rangle$ serves as an infinitely long tape with finite input, and $|x\rangle$ corresponds to the tape location being currently scanned.

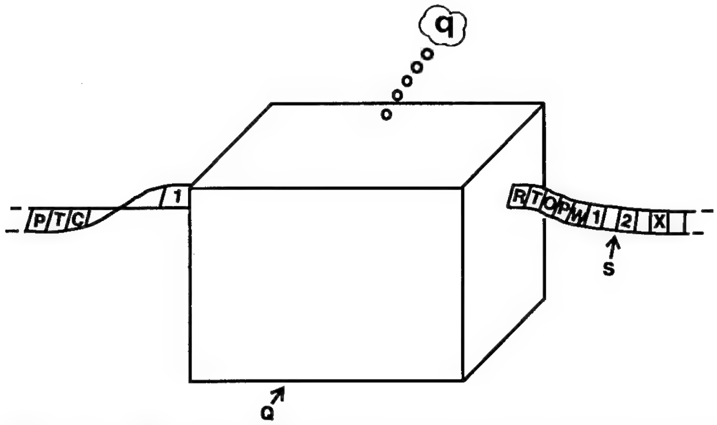


Figure 1. *The Turing Machine (TM) (named after its innovator, A. M. Turing) is a formalization of the idea of an effective process; i.e., algorithm. It can be pictured as a box with a tape running through it. The tape consists of a sequential collection of squares that may extend indefinitely in either direction. The box can be in any one of the internal states contained in a finite set, Q, and is able to scan or print on the tape any symbol in a finite set, S. The machine is started by being set to scan some tape square while in initial internal state* q_0. *The action of the machine is determined by a partial transition function* $\theta: Q \times S \rightarrow Q \times S \times \{L,R\}$, *where L(R) means shift the tape one square to the left (or right). The machine may continue working indefinitely or may eventually stop when the internal state and symbol scanned form a pair not in the domain of* θ. *Turing showed that there exists a Universal Turing Machine (UTM) that is capable of simulating any TM.*

The dynamics of a DM are generated by a constant unitary operator, $\hat{U}$, which specifies the evolution of any state in $\mathscr{H}$ during a single computation step of duration Δt. Thus, $|\psi(k\Delta t) = \hat{U}^k|\psi(0)\rangle$, where k is a nonnegative integer, $|\psi(0)\rangle = \Sigma a_{\bar{m}}|0;\bar{0};\bar{m}\rangle$ with a finite number of the $a_{\bar{m}} \neq 0$, and $\Sigma|a_{\bar{m}}|^2 = 1$. Here $|\bar{0}\rangle = |0,...,0\rangle$, $|\bar{m}\rangle$ contains the input, and $a_{\bar{m}}$ vanishes when an infinite number of $m_i = 1$ in $\bar{m}$. The matrix elements of $\hat{U}$ are constrained to the form:

$$\langle x';\bar{n}';\bar{m}'|\hat{U}|x;\bar{n};\bar{m}\rangle = [\hat{U}^+(\bar{n}',m'_x|\bar{n},m_x)\delta_{x',x+1} + \hat{U}^-(n',m'_x|\bar{n},m_x)\delta_{x',x-1}]\Delta.$$

Here, $\delta_{x',x\pm1}$ ensures a unit change in tape position, and $\Delta = \prod_{y\neq x}\delta_{m'_y,m_y}$ constrains memory involvement to location x during a computational step. $\hat{U}^\pm$ are arbitrary functions that are consistent with the unitarity of $\hat{U}$ and describe the dynamical motion. There is, thus, a DM for every permitted choice of $\hat{U}^\pm$ and each exists, i.e., there is a quantum computation for a given input $|\bar{m}\rangle$, if its run time expectation value is finite. The observable $\hat{n}_0$ can serve as a completion flag and can be internally set to unity if the associated DM exists.

TMs are DMs that are in an $|x;\bar{n};\bar{m}\rangle$ basis state at the end of each computational step. The $\hat{U}^\pm$s for DMs that are equivalent to (reversible) TMs are given by:

$$\hat{U}^\pm(\bar{n}', m'|\bar{n},m) = \tfrac{1}{2}\left(\delta_{\upsilon,\bar{n}'}\delta_{\mu,m'}[1\pm\gamma]\right),$$

where $\upsilon:(\bar{n},m)\rightarrow\{0,1\}^M, \mu:(\bar{n},m)\rightarrow\{0,1\}$, and $\gamma:(\bar{n}, m)\rightarrow\{\pm1\}$.

The universal quantum computer (UDM) not only subsumes the properties of the UTM, but also simulates with arbitrary precision any quantum computer. It does this by permitting the utilization of eight distinguished instruction sets that provide the following four unitary transformations and their inverses for the evolution of single computational binary basis states, i.e., $|0\rangle$ and $|1\rangle$, into linear superpositions:

$$\begin{pmatrix}\cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi\end{pmatrix}; \begin{pmatrix}\cos\varphi & i\sin\varphi \\ i\sin\varphi & \cos\varphi\end{pmatrix}; \begin{pmatrix}e^{i\varphi} & 0 \\ 0 & 1\end{pmatrix}; \begin{pmatrix}1 & 0 \\ 0 & e^{i\varphi}\end{pmatrix}.$$

Here, φ is some irrational multiple of π. These transformations generate under the operation of composition a group G' that is dense in the group G of all unitary transformations on the Hilbert space spanned by $\{|0\rangle, |1\rangle\}$; i.e., $G'\subset G$ and every open subset of G contains elements of G'. Thus, desired transformations of individually specified binary states can be produced with arbitrary precision via generator composition using catenations of these distinguished instruction sets. Indeed, there are instruction sets that induce analogous evolutions for finite numbers of such states. Hence, unlike the UTM, the UDM can employ the quantum mechanical property of state superposition to

provide massive parallel processing capabilities. Although variants of Deutsch's model can be found in recent literature (see below), their features still resemble those of the UDM.

Quantum Complexity Theory

Answers to fundamental questions concerning the computational power of quantum computing devices will ultimately be obtained from the study of quantum complexity theory (QCT). QCT is concerned with the characterization and classification of the intrinsic computational difficulty—i.e., time and/or space resource requirements defined in terms of the properties of the UDM or its variants—in obtaining a solution to a problem.

While it is known that the class of functions computable by the UDM is the same as that computable by the UTM, because of the newness of QCT, little is known about quantum complexity classes and their relationships to classical complexity classes (those defined in terms of the UTM or its variants (see Figure 2)). Thus, the question of whether harnessing quantum phenomena for use in computational processes provides enhanced computational power has not been satisfactorily answered. However, recent research has strongly suggested that this may indeed be the case. Deutsch and Josza[9] constructed a problem that could be solved exponentially faster when using a quantum computer rather than a classical computer. Bernstein and Vazirani[10] established that a significant speedup is possible for certain classes of problems by exhibiting an oracle problem that can be solved in polynomial time on a quantum computer but requires superpolynomial time on a classical computer.

Shor[11] has built upon these results to provide the first real indication of the intrinsic computational power of quantum computers. He has developed an algorithm for factoring integers on a quantum computer in polynomial time. Heretofore, no polynomial time-factoring algorithm was known, and the factoring problem was believed to be so difficult that encryption systems were based upon it. Thus, if a quantum computer could be built, codes that require months or years to crack using a suite of contemporary classical computers could be cracked in seconds using a quantum factoring device.

Another interesting result that illustrates the potential power of quantum computers has recently been reported by Cerny´.[12] He describes a physically permissible quantum computer that uses quantum parallelism to solve in polynomial time the Traveling Salesman Problem (TSP), a well-known NP-complete problem (see Figure 3). A particle traverses the computer to provide in polynomial time a resultant state that enumerates all possible tours in the form of that superposition of states, with a permitted route and its distance encoded in each state. Thus the particle "knows" the solution in polynomial time.

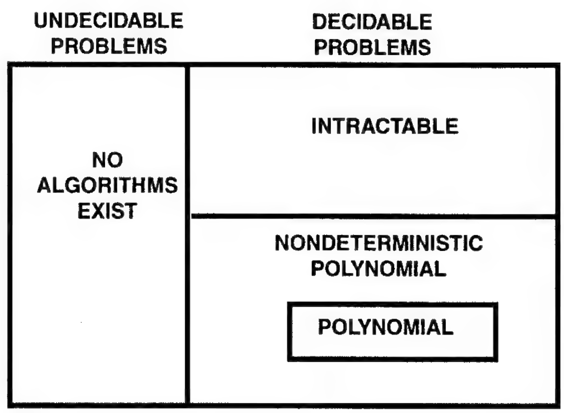


Figure 2. *The complexity of a problem is a fundamental invariant of computation and is independent of the particular algorithm used. Classical complexity theory is defined in terms of the TM model. An algorithm is of time complexity t(n) if the computation performed by the associated TM requires at most t(n) moves for all input words no longer than n. If g(n) is a polynomial function, then an algorithm is a polynomial time algorithm if there is a constant, c, such that* $|t(n)| \le c|g(n)|$ *for all* $n \ge 0$. *Otherwise, it is said to be an exponential time algorithm. Using these notions, problems may be classified as shown above:*

- *Undecidable problems are those for which no algorithms exist.*
- *Intractable problems are those that can only be solved using exponential algorithms.*
- *Nondeterministic polynomial (NP) problems are those that can be solved in polynomial time if the computational path that should be followed can be correctly guessed.*
- *Polynomial (P) problems are those that can be solved in polynomial time. (A problem is considered to be well solved only if a P algorithm is known for it.)*

Difficulty arises when the observer wishes to also "know" the solution.

Cerny´ uses his computer to make an important point concerning this issue; i.e., the connection between the complexity of a problem and the observation of its solution. A readout measurement is made using a Stern-Gerlach device that separates states according to the length of the route. This, of course, "collapses" the superposed state into one of its constituent route/distance states with an associated probability, p, inversely proportional to the number of permitted routes. Clearly, an ensemble of particles can be used to find the minimum distance route. However, since p is extremely small for large TSPs, an "exponentially large" number of particles must be used to observe the minimum distance route. As a result of this, Cerny´ conjectures the existence of a complementarity principle concerning the time and energy needed to perform an NP-complete computation. While we will not pursue it further

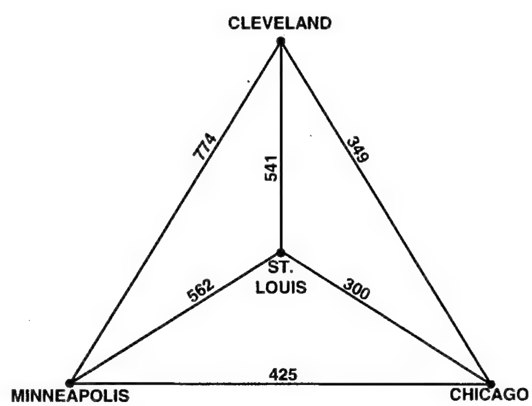


Figure 3. *The problem of finding the shortest route that visits each of a given collection of cities and, finally, returns to the city of origin is called the Traveling Salesman Problem (TSP). If Chicago is the city of origin in the simple example above, then the shortest route is Chicago, Minneapolis, St. Louis, Cleveland, and Chicago, with a distance of 1,877 miles. This general problem is one of the class of nondeterministic polynomial (NP)-complete problems known to contain hundreds of different problems notorious for their computational intractability. NP-complete problems have two important properties: (1) if any NP-complete problem is solved by an efficient algorithm, then all of them are; and (2) all algorithms currently used (in the classical sense) can always "blow up" exponentially. It is believed (but not yet proven) that NP-complete problems have no efficient solutions.*

here, this conjecture dovetails quite well with an extension of the Church-Turing thesis enunciated by Deutsch, which establishes an equivalence between physics and computer science.

Quantum Cryptography and Teleportation

Let us now briefly direct our attention to two rapidly evolving technologies that will likely impact information processing in the near future—quantum cryptography and quantum teleportation. The former is a technologically feasible approach to using purely quantum mechanical effects for a specific information processing purpose. It is a technique that allows the distribution of key data (secret random sequences of bits used for message decoding) in a manner that "guarantees" its secrecy. While several distinct methods have been reported in the literature, they all are characterized by reliance upon a quantum effect (i.e., the uncertainty principle, Bell's inequalities, and properties of nonorthogonal states) and a protocol that exploits this effect to "guarantee" secrecy.[13-17] (The word guarantee has been enclosed in quotes because it has not yet been proven that the associated protocols are totally secure.)[18] Several prototype quantum cryptography systems have been constructed that are capable of sending key data over short distances.[19,20]

A fascinating discovery has recently been made by Bennett, et al.[21] They have shown how it is theoretically possible to teleport information using the Einstein-Podolsky-Rosen (EPR) effect. Such teleportation is based upon the distinction between classical information (which can be duplicated freely, is undisturbed by observation, and has a transmission speed limited by the speed of light) and quantum information (which can't be readily duplicated, is disturbed by observation, and appears to be instantaneously transmitted under certain conditions). In particular, an unknown quantum state can be partitioned into classical and quantum information parts that can be sent through separate channels and reassembled to obtain the original state. Although the quantum information is transferred instantaneously via EPR correlations, the classical information required to exactly reconstruct the state must be transferred using a conventional medium. Thus, the entire process still requires a finite amount of time and violates no physical

laws. It is easy to see that such a scheme could be very useful for teleporting quantum information over great distances, without concern for the associated degrading effects of attenuation and noise. While quantum teleportation is currently not technologically feasible from an operational perspective, it is interesting to note that a realizable quantum teleportation machine based upon the notions of Bennett et al., which will teleport atomic states, has been proposed by Davidovich, et al.[22] Hence, we see the rapidity with which progress is being made.

Concluding Remarks

Although the basic principles of quantum computation theory were established by Deutsch nearly a decade ago, its implications are not yet fully understood and are likely to be far-reaching. Indeed, only recently is it beginning to have an influence upon both the computer science and physics communities.

Quantum computers do not yet exist. Nonetheless, some researchers are beginning to produce design concepts that might eventually evolve into working models.[23-25] Despite the significant technological barriers that must be overcome, revolutionary advances in micro- and nano-technologies, as well as the persistent demands of the information age, will likely converge to ensure the ultimate maturity of quantum computing devices.

References

1. Benioff, P., "The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines," *Journal of Statistical Physics, Vol.* 22, 1980, p. 563.
2. Benioff, P., "Quantum Mechanical Hamiltonian Models of Turing Machines," *Journal of Statistical Physics*, Vol. 29, 1982, p. 515.
3. Benioff, P., "Quantum Mechanical Hamiltonian Models of Discrete Processes that Erase Their Own Histories: Application to Turing Machines," *International Journal of Theoretical Physics*, Vol. 21, 1982, p. 177.

Glossary

Bell's Inequality - An experimentally verifiable relationship between quantum mechanical correlation functions that suggests the existence of nonlocal interactions between spatially separated particles.

Church-Turing Thesis - "A Problem is computable if, and only if, it can be computed by the UTM." As a working assumption this makes a formal theory of computability possible, because it implies the existence of a well-defined boundary between that which is decidable (computable) and that which is not.

Eigenvalue/Eigenvector - A scalar λ is an eigenvalue, and a nonzero vector $|\Psi\rangle$ is an eigenvector of a linear operator $\hat{A}$ if $\hat{A}|\Psi\rangle = \lambda|\Psi\rangle$. In quantum mechanics, observables in nature are represented by linear operators, physical states are represented by eigenvectors, and the result of a measurement of an observable is one if its eigenvalues.

Einstein-Podolsky-Rosen Effect - The possibility that a measurement performed in one place on one member of a quantum pair of particles can instantaneously influence in a specific way the other member located at an arbitrary distance from the original particle.

Group - A set and an associative binary operation *, along with an element i, such that i * x = x for all elements x and, for every x, there is a y in the set with y * x = i.

Hamiltonian - The energy operator for a quantum mechanical system.

Hilbert Space - The mathematical setting for quantum physics in which physical states are vectors, and physical observables are operators in the associated space.

Linear Superposition of States - The mathematical notion that a quantum system can simultaneously exist in more than one state. A measurement process upon such a superposition collapses the system into one of its constituent states.

Oracle - A "subroutine" that, when plugged into a TM or a DM, always provides cost-free correct answers to yes/no questions asked of it.

Stern-Gerlach Device - A measurement apparatus that separates an ensemble of quantum particles according to their quantum state.

Uncertainty Principle - The doctrine that provides limits upon how accurately two quantum mechanical observables may be simultaneously measured.

Unitary Operator - An operator that has an inverse equal to the adjoint of the operator. The evolution of an undisturbed quantum mechanical system is deterministic and is described by a unitary operator.

4. Benioff, P., "Quantum Mechanical Hamiltonian Models of Computers," in *New Techniques and Ideas in Quantum Measurement Theory*, Boland, B.; Culliman, J.; and Malmoli, S. (eds.); New York Academy of Sciences, New York, 1986, p. 475.
5. Peres, A., "Reversible Logic and Quantum Computers," *Physical Review A*, Vol. 32, 1985, p. 3266.
6. Feynman, R., "Simulating Physics with Computers," *International Journal of Theoretical Physics*, Vol. 21, 1982, p. 467.
7. Feynman, R., "Quantum Mechanical Computers," *Foundations of Physics*, Vol. 16, 1986, p. 507.
8. Deutsch, D., "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer," in *Proceedings of the Royal Society of London: A*, Vol. 400, 1985. p. 97.
9. Deutsch, D. and Josza, R., "Rapid Solution of Problems by Quantum Computation," in *Proceedings of the Royal Society of London: A*, Vol. 439, 1992. p. 553.
10. Bernstein, E. and Vazirani, U., "Quantum Complexity Theory," in *Proceedings of the 25th ACM Symposium on the Theory of Computation*, 1993, p. 11.
11. Shor, P., "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, IEEE Press (to appear), 1994.
12. Cerny´, V., "Quantum Computers and Intractable (NP-complete) Computing Problems," *Physical Review A*, Vol. 48, 1993, p. 116.
13. Bennett, C. H.; Brassard, G.; Breidbart, S.; and Wiesner, S., "Quantum Cryptography, or Unforgeable Subway Tokens," in *Advances in Cryptology: Proceedings of Crypto 82*, 1982, p. 267.
14. Ekert, A. K., "Quantum Cryptography Based on Bell's Theorem," *Physical Review Letters*, Vol. 67, 1991, p. 661.
15. Collins, G. P., "Quantum Cryptography Defies Eavesdropping," *Physics Today*, Vol. 45, 1992, p. 21.
16. Bennett, C. H.; Brassard, G.; and Mermin, N. D., "Quantum Cryptography Without Bell's Theorem," *Physical Review Letters*, Vol. 68, 1992, p. 557.
17. Ekert, A. K., "Quantum Keys for Keeping Secrets," *New Scientist*, Vol. 139, 1993, p. 24.
18. Berthiaume, A. and Brassard G., "The Quantum Challenge to Structural Complexity Theory," in *Proceedings of the Seventh Annual Structure in Complexity Theory Conference*, 1992, p. 132.
19. Bennett, C. H.; Bessette, F.; Brassard, G.; Salvail, L.; and Smolin, J., "Experimental Quantum Cryptography," *Journal of Cryptography*, Vol. 5, 1992, p. 3.
20. Ekert, A. K.; Rarity, J. G.; Tapster, P. R.; and Palma, G. M., "Practical Quantum Cryptography Based on Two-Photon Interferometry," *Physical Review Letters*, Vol. 69, 1992, p. 1293.
21. Bennett, C. H.; Brassard, G.; Crepeau, C.; Josza, R.; Peres, A.; and Wootters, W. K., "Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolosky-Rosen Channels," *Physical Review Letters*, Vol. 70, 1993, p. 1895.
22. Davidovich, L.; Zagury, N.; Brune, M.; Raimond, J. M.; and Haroche, S., "Teleportation of an Atomic State Between Two Cavities Using Nonlocal Microwave Fields," *Physical Review A*, Vol. 50, 1994, p. R895.
23. Caulfield, J. H. and Shamir, J., "Wave-Particle-Duality-Processors: Characteristics, Requirements, and Applications," *Journal of the Optical Society of America*, Vol. 7, 1990, p. 1314.
24. Lloyd, S., "A Potentially Realizable Quantum Computer," *Science*, Vol. 261, 1993, p. 1569.
25. Lloyd, S., "Envisioning a Quantum Supercomputer," *Science*, Vol. 263, 1994, p. 695.

The Author

ALLEN D. PARKS is a Principal Astronomer in the Advanced Computation Technology Group of the Systems Research and Technology Department. He received a B.A. degree in mathematics from Marshall University in 1969 and a Ph.D. in astronomy from the Pennsylvania State University in 1975. Prior to joining the Naval Surface Warfare Center, Dahlgren Division in 1981, he was employed by the General Dynamics Corporation, Computer Sciences Corporation, and the General Electric Company. His areas of interest include the theory of computation, database theory, quantum mechanics, graph theory, and the theory of semigroups. He has been recognized for his work in space systems design and development by the Defense Mapping Agency, has been awarded the Navy Meritorious Civilian Service Medal, and has received excellence awards for independent research and exploratory development. In addition, he has authored over 50 technical publications.

Time Series Analysis from a Chaotic Point of View

Robert Cawley, Guan-Hsong Hsu, and Liming W. Salvino

Nonlinear dynamics, "chaos theory" for short, is a broad discipline having applications in nearly every field of science and engineering. It is a field driven by mathematics and experiment at the same time. Its mathematical side is dynamical systems theory, a field of work in a sense akin to that of integral calculus. It is not surprising that its applications should be ubiquitous. In the early 1980s, the Navy was first in the Department of Defense (DoD) to realize the long-term importance of this new field, and the White Oak Detachment of the Naval Surface Warfare Center, Dahlgren Division (NSWCDD) was surely among the earliest to develop across-the-board expertise in it. It was clear at the outset that the analysis of experimental and field data from a dynamical systems perspective must be an important area of investigation and, although the research carried out at NSWCDD has been quite diverse, chaotic data analysis remains a significant part of that work. One of the most difficult issues confronted by the chaos community has been that of learning how to make correct chaos/no-chaos calls in noisy experimental data. Time-series analysis methods that we have developed for dealing with this problem are being applied to a variety of problems of Navy interest.

Introduction

Simple nonlinear dynamics govern the behavior of many experimental systems in controlled laboratory conditions, and reliable observations of chaos abound. The documentation of this fact is all through the scientific literature of the last 15 years. There are even theoretical predictions of chaos for bodies orbiting in black-hole systems. Machining and welding automation are two examples of technology settings where simple nonlinear dynamical processes, possibly masked by high noise, may exist and be exploitable.

For instance, in the case of tool workpiece interaction, little is actually known about whether operating conditions exist that produce chaos. But chaos seems likely to occur in the important areas of high-speed machining, hard turning, and situations of flexible-tool workpiece interaction. Professor Frank Moon of Cornell University has speculated that chaotic machining in shallow cuts may produce smooth finishes. Another possibility may be to create a controlled surface roughness (where this may be desirable) by means of chaos-induced chatter.

The physics of welding is more complicated, due to weld material transfer involving magnetohydrodynamic (MHD) and plasma processes. However, it is also a droplet process, which may have universal scaling regime properties, including simple nonlinear dynamics, as has been seen experimentally in the dripping faucet.

In each of these manufacturing technology areas, chaos-theory-based system diagnostics might be employed to prevent unwanted behavior, such as bad welds, say; or online time series forecasting might be used for monitoring and real-time control. These methods do not require the existence of low-dimensional chaos to work.

Finally, very broadly, it is standard engineering practice to avoid nasty nonlinear effects where possible; e.g., introduce damping, or design away from risky regions of parameter space. In most cases this is surely the right thing to do, but in some it may be that opportunities are lost by not considering nonlinear options. In addition, affordability may sometimes be a driver in that cost savings may be had by seeking the more efficient designs that can occur when one pushes into more highly driven nonlinear regimes.

Biting the Bullet

To determine whether in a real system there are nonlinear dynamics operating, or possibly other processes exploitable by chaos theory methods, we must bite the bullet of chaotic data analysis. There is opportunity here; for even though a system behavior may be extremely complicated, enhanced predictability might be possible, together with possibilities for control.

However, to address the problem of analyzing real experimental time-series data, it is necessary also to deal with the problem of noise. To detect, characterize, predict and control chaos when it is there, we clearly need to know how chaos differs from noise. We begin with a bit of an essay on where chaos comes from, what it is, and what it is not, and we conclude with a description of a thorny contemporary problem. Then we describe methods we have developed here in the Dahlgren Division to deal with this problem, in order to accomplish the first and most difficult step—to detect reliably the presence of dynamics in real-time series data.

In fact, in this article we describe a general, systematic method for assessing the presence or absence of determinism in time series. We highlight two inherently interactive key features of our approach that conspire to make this treatment promising and, so far, fully successful: the use of a smoothness detector and the use of chaotic noise reduction.

We remark right here at the outset that chaos is deterministic, and randomness is not deterministic. This is an inherent part of our subject. Later we shall say exactly what we mean by deterministic.

Where Do Nonlinear Dynamics Come From?

The roots of engineering technology go back a long way. Those of nonlinear dynamics are more recent; they begin with Isaac Newton. We will tell some of this story because it provides a vehicle for shedding light on the subject of chaos theory itself.

Differential calculus was invented three centuries ago,[1] and along with it the harder problems of integral calculus, of finding functions specified by giving only derivatives, or only slightly more generally—but usually a lot harder—relations among two or more derivatives. The same Isaac Newton who invented calculus also formulated the correct physics for the dynamical laws of motion: $F = ma$.

With these two strokes, Newton bequeathed to posterity an enormous range of usually very hard problems, problems of integration—integration to find the motions specified by the expressions of the dynamical laws as relations among the fundamental observables of those motions: the positions, velocities, and accelerations. We call these relations differential equations. Each unique physical system had its own version of those laws, and with each came another sometimes easy, but more often than admitted in textbooks, hard mathematics problem of integration.

The physical systems governed by Newton's laws of motion span a litany of topics. The motions of the planets and moon, of bodies connected by springs, vibrating beams, complex swirlings of fluids in a laboratory experiment, or of the ocean and atmosphere are just a few.

All these are governed by differential equations. Today we can add many more, involving reaction rates among constituents in chemical mixtures, nerve impulse propagation in biological systems, light propagation in optical laser cavities, the motions of bodies orbiting in multi-black-hole spacetimes, and just about any kind of real-world classical system properly modeled by relations among physical observables and their derivatives; that is to say, by systems of differential equations.

In the language of mathematics, a system of differential equations is a model for a "dynamical system."

A dynamical system governing a real-world system is normally derived by application of fundamental laws, such as those of mechanics,

local thermodynamic equilibrium, electromagnetic theory, and the like, to the actual system at hand, or rather, more often, to a simplified model of the actual system. When a physical system is complex, the importance of modeling may become paramount, simply to be able to have a picture we can comprehend and hope to keep track of. Models promote understanding, we verbalize the mathematics expressing them, and those words become promoted to physical concepts.

But some physical, chemical, biological, and engineering systems may resist modeling efforts, making the governing, or underlying dynamical systems that describe them, defy the imaginative efforts of theorists to find a tractable picture, a physical understanding. An apocryphally familiar crucible of examples is the weather, where the laws and governing equations are pretty well-known, but the solution is not. And chaos is about solutions, as we shall see.

The New Calculus

For two centuries, our scientific and engineering forebears approached the deeply fundamental problem of integration, of solving differential equations, of quite literally getting the *answers*, (note that experiment gives answers) by hit and miss trickery, or by brute force. Out of this approach came the classical elementary functions of calculus, such as elliptic integrals, Bessel functions, and hypergeometric functions of all sorts, not to mention nonlinear equations and systems having proper names attached, like the Riccati equation and Kepler system, and even the whole framework of classical perturbation analysis based on Taylor series. This centuries-long effort did not predict chaos, but only simpler things like neat elliptical orbits, the open rosettes of precessing orbits, Lissajous figures and, most pervasively, harmonic motions and their cousins, the higher harmonics.

But about one century ago, an ingenious shift in *approach* was introduced by the French mathematician, Henri Poincaré.[2] Instead of attempting to solve given systems of differential equations he, in effect, reframed the problem as one of "what are some of the properties of the solutions that can occur?" It's a little like, "What can there be in the world?" Like Newton before him, whose ruminations about Nature had led him to invent the calculus, Poincaré was led to topology, and to the formulation of the first foundations of modern dynamical systems theory. Poincaré's immediate intellectual successor, the American mathematician, George David Birkhoff, expanded fundamentally on the approach of Poincaré and carried the program further.[3] But the consequences of the early work of Poincaré and Birkhoff entailed such horrendous complexities and were so discouraging that research into dynamical systems came to a virtual halt for nearly half a century!

This shift in approach to the old calculus problem of integration, the shift away from getting the answers, engineered by Poincaré and Birkhoff, the shift from that of a classical algebra-like analysis to a qualitative geometrical, or topological, approach to characterizing logically possible solutions, deserves to be called a paradigm shift. This near stillbirth was revived by mathematicians in the 1960s,[4,5] much closer to a time when the computer would become available as a catalyst to assist with the monstrous mathematical intricacies uncovered earlier. The new paradigm was on its way to becoming a literally new calculus.

What Is Chaos?

What we have said so far is something like this: calculus is everywhere in the scientific and engineering world, and calculus doesn't give the answers even though we may know the governing laws. It gives problems, systems of differential equations typically, that cry out for integration. Then we said that Poincaré and Birkhoff cheated by "going to the back of the book" to see what kinds of answers there are, only they couldn't read some of them very well because the answers are sometimes very complicated. We mentioned the computer, a device that nowadays has little trouble displaying complicated answers. Finally, we said the new paradigm, the "view from the back of the book," constitutes a veritable new calculus. We are nearly ready now to say what chaos is.

In the back of the book, where the answers are, instead of differential equations, and actually also their close cousins, the difference equations (or, equivalently, maps), one finds dynamical systems displayed as flows and maps on manifolds.

The manifold for the dynamics is called the phase space, and it "shows" the answers graphically; i.e., geometrically. A system of ordinary differential equations will typically possess trivial constant solutions and periodic (limit cycle) solutions. (Externally driven systems, whose corresponding vector fields cannot vanish, are an exception.) The constant solution shows up as a fixed point in the manifold for the corresponding flow, while the sustained *regular* oscillation corresponding to the periodic solution appears as a simple closed curve. Owing to the geometric character of this phase-space picture, periodic solutions are also called periodic orbits, with the system point cycling endlessly around the curve.

But another kind of solution that we now know to occur typically in systems of differential equations, and which often used to be discarded when it was turned out by a computer (a well-known exception is E. N. Lorenz, who did not file his ugly duckling in a bottom drawer),[6] instead is simply aperiodic; that is, a sustained *irregular* oscillation. An aperiodic solution rendered in the phase space by a computer gives what looks like a pretty good fractal.[7]

This kind of solution behavior is called chaos, and the phase-space orbit, a chaotic orbit. An oft-cited example is the Lorenz system specified by the following equations in three-dimensional space, $\mathbb{R}^3$,

$$\begin{aligned} \frac{dx}{dt} &= \sigma (y - x) \\ \frac{dy}{dt} &= \rho x - y - x z, \qquad (x, y, z) \in \mathbb{R}^3 \qquad (1) \\ \frac{dz}{dt} &= -\beta z + xy \end{aligned}$$

where σ, β, and ρ are constants. For the parameter choice, $\sigma = 10$, $\beta = 10/3$, and $\rho = 28$, the fractal of Figure 1 results.

There are issues we can't go into here due to lack of space. For instance, in a mathematics

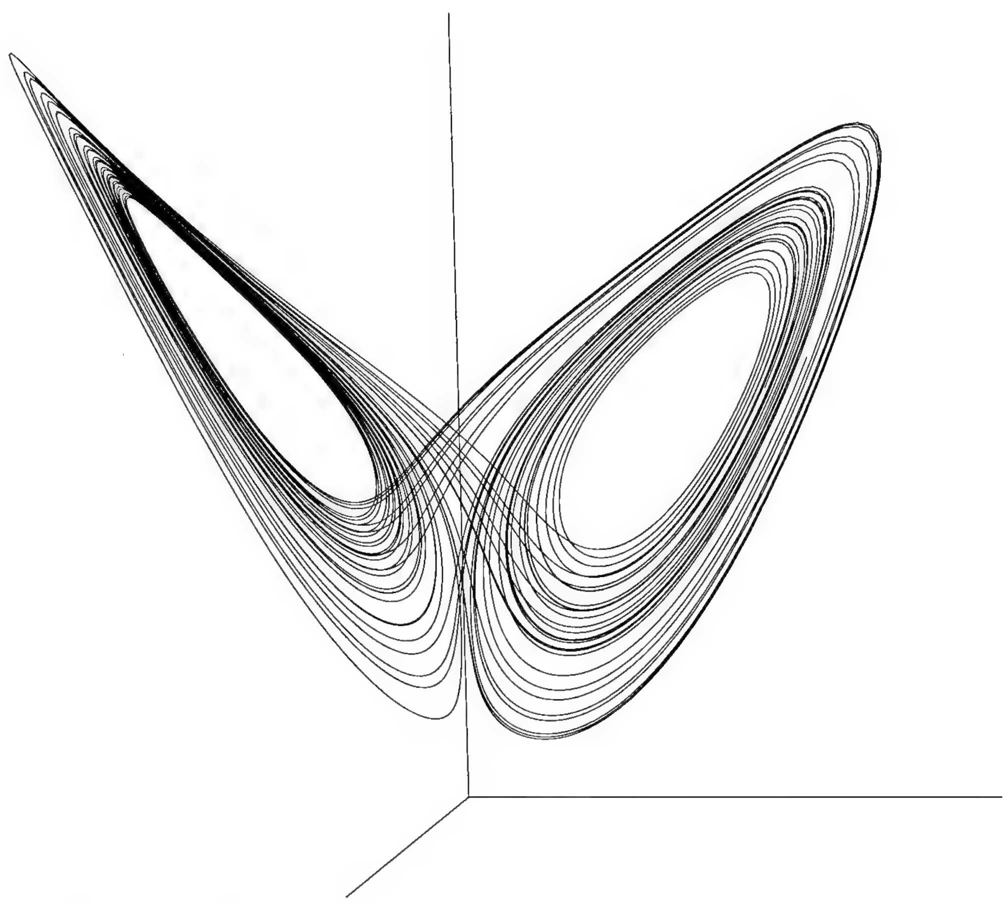


Figure 1. *Fractal issuing from the Lorenz system. The trajectory of (x(t), y(t), z(t)) quickly settles down to motion along the complicated curve shown.*

setting, we should insist upon a precise definition of chaos, so we can really say something about it. Thus we might require there be a "Smale horseshoe"[5] in the dynamics. At another level of refinement, we may want to require the existence of at least one positive Lyapounov exponent in order to distinguish chaotic from strange nonchaotic behavior. In the latter, the phase portrait may be a fractal, but a time series issuing from the equations of motion, although aperiodic, doesn't possess sensitivity to initial conditions; i.e., exponential mean rates of separation of initial conditions in phase space, a feature usually required of chaos. (We note in passing that the first experimental observation of strange nonchaotic behavior in a nontrivial physical system was reported by NSWCDD.[8] The experiment was the second in a series of three collaborations between the Nonlinear Dynamics Group and the Magnetism Group at White Oak.)

What the Science of Chaos Isn't

The science of chaos isn't physics; it's a kind of kinematics. It isn't physics, or chemistry, or biology, or meteorology, or mechanics, or any of the scientific or engineering areas in which chaotic behavior may be found. Like periodic oscillation, chaos is a universal category of behavior. The science is really that of dynamical systems theory, together with its applications and manifestations in the world.

If we find that a noisy aperiodic quantity is actually the output of a chaotically behaving system, we can infer the likely existence of a simple set of rules producing that behavior. And if we know that much, we can be encouraged to search for a simple model despite extremely complicated behavior in the data, and where otherwise we might only have had a missed opportunity. Thus we have to put physics in to get physics out, which, as long as we are not looking for a "silver bullet," is just fine.

The experimentalist surely operates from the *back* of the book. He measures answers, right answers generally, and the scientific enterprise is to learn from these as much as possible about the questions. These reflect the underlying laws and principles of physical behavior, or the succinct expressions of possible simple models. The fact that a discipline as intricate and abstract as topology should lie bedfellow "in the back of the book" with something as down to earth as experimental science seems to us an extraordinary irony.

Experimental results always will be contaminated by some noisy background interference, however, so one really measures distorted versions of the answers. While the presence of noise need not necessarily mask detection of a periodic system oscillation, it can create significant problems in an experimentalist's ability to recognize chaotic behavior. As a contrasting for instance, chaotic time series have infinite bandwidth, a complication not too troublesome for periodic behavior. Noise contamination can also limit the efficacy of experimental control of chaos.[9]

A Crucial Major Advance in Experimental Data Analysis

A crucial major theoretical advance linking dynamical systems theory and experimental science was made about fifteen years ago; it is called the delay coordinate construction (DCC), or Ruelle-Takens construction, and is apparently due to David Ruelle.[10,11] It permits the reconstruction of a valid phase portrait for the orbit of a differentiable dynamical system from knowledge of only a single measured quantity. Thus one does not need to measure time series of all the active degrees of freedom to get at the phase-space physics. This basic theorem is due to Takens, and it makes an experimental science of chaos possible. This key idea is depicted in detail in Figure 2 (on facing page).

The False-Alarm Problem

The idea that extremely complicated data may sometimes reflect an underlying simplicity is an exciting and seductive one. With the help of Takens' theorem and a few other techniques, there have been some nice experimental successes. But there has been, and still is, a false-alarm problem. For, unhappily, this seductive idea also has been overindulged on occasion. Some of the claims for experimental observation of chaos are surely erroneous.

The problem is not Takens' theorem or the DCC, however. A common trap into which researchers have fallen has been to assume the presence of nonlinear dynamics and then measure an observable of choice, such as a fractal dimension or Lyapounov exponent for a phase portrait. A practical error has been to fail to control for disastrous effects of noise. The logical error is obvious since any time series can be represented by a phase portrait; but other options have not seemed to be available. For a brief history of chaotic time series analysis, see Reference 12.

The derivative idea that chaotic dynamics might be uncovered after removal of a contaminating noisy background is an easy extension of this program, and possibly even more exciting and seductive. So far, it has not captured the imagination of the scientific community, perhaps owing to the complexity of the chaotic noise reduction process. It is probably not useless to note that this approach, too, is likely to be abused in time, and there will be another false alarm problem. But if we are careful and if we are lucky, we will also find some successes here.

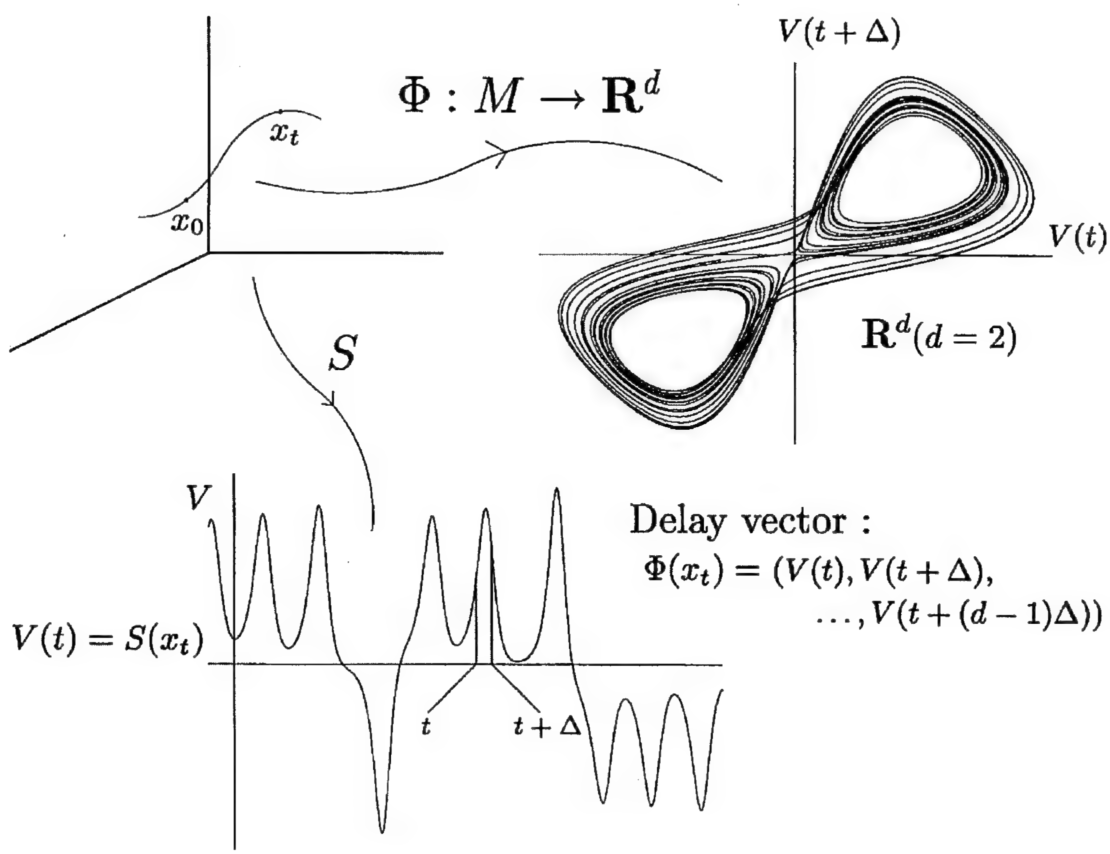


Figure 2. *Ruelle-Takens embedding construction for the Lorenz system. For this system, the manifold, M, for the dynamics is just the Euclidean three-space: $M = \mathbb{R}^3$. x_t represents the point at time t from the curve shown in Figure 1. The measurement mapping, S, from points x_t to the real numbers, models the connection between the system under observation, (here a flow in $\mathbb{R}^3$) and the experimental observation itself, V(t), which is typically an instrument reading, such as a voltage. The delay construction on V(t) is a time-dependent vector of d components, shown here for the case of embedding dimension d = 2. It is not a true embedding in this example as the Lorenz system is three dimensional. This is reflected in the evident occasional self-intersections ("imperfections") of the fractal curve. If d is chosen large enough, the self-intersections will be absent, giving a true embedding, i.e. a "perfect" representation. The d = 2 picture can also be considered as a projection into the first two components of a true embedding (vid. Figure 1). Takens' theorem, an application of the Whitney embedding theorem, asserts that, with respect to smooth (C^2) flows on differentiable manifolds, M, and mappings, S, the delay coordinate construction is generically an embedding, Φ, of M into $\mathbb{R}^d$ if $d > 2m$, where m is the dimension of M.*

Thus, historically there has been and, in the research community at large, still is a problem—that of being fooled with false alarms. We turn now to our two-prong approach to this problem: an algorithm for detecting determinism and an algorithm for chaotic noise reduction. These two inherently interactive elements conspire to make our treatment promising and, so far, successful.

Smoothness Implies Determinism

Before we begin, we need to deliver a *caveat* or two. Some deterministic systems may be high dimensional and look pretty noisy. If they have too many degrees of freedom, it may be necessary, or even best, to regard them as effectively random for all practical purposes. Indeed, the effort to use the data analysis methods developed for nonlinear dynamics makes the best sense when the system is effectively low dimensional.

In addition, as a practical matter, any test for determinism, any time series measurements, or dynamical analysis, including ours, is necessarily performed in relation to some, generally low, chosen dimension, such as may be used to construct the phase portraits. Nevertheless, as a matter of principle, and even though in some cases the point may be just an academic one, any conclusion asserting the absence of determinism is *inherently limited by this chosen dimension*, for a finite (higher) dimensional determinism yet might be present. This said, we may now proceed.

The rest of the material in this section is an explication of Reference 13. It was presaged by earlier work by Kaplan and Glass[14, 15] and Wayland, et al.[16]

Uniqueness Theorem for Solutions to Differential Equations

It is actually a fact that smoothness in phase space implies determinism in time series. The mathematical situation is this: chaotic behavior is produced by nonlinear ordinary differential equations and maps on manifolds. As long as the right-hand side of a system of ordinary differential equations is a smooth (i.e., locally Lipshitz) function of position, its solutions are uniquely fixed from any given initial condition, and nearby points on the phase space behave similarly under time evolution. These continuity properties thus imply unique future behavior, that is, *smoothness implies determinism*. (For maps, see Reference 13.)

A flow that is only C^0 need not evolve uniquely (deterministically) from a given initial condition. A simple one-dimensional example of a non-deterministic C^0 system is $\dot{x} = (1 - x^2)^{1/2}$, $x(0) = 0$.

Thus, sufficient continuity on an embedded phase space is enough to imply determinism in time series. (This result is as strong as we have put the matter. If smoothness on the embedded phase space is established, the existence of a smooth dynamical system in the system producing the dataset and, therewith, of deterministic evolution, is also.) Moreover, as we show shortly, it is possible to define infinitely many arbitrary vector fields over a phase portrait for a time series. We exploit this arbitrariness to generate a detector of smoothness and, therefore, of determinism in time series.

Our Method

Our method is simple and easy to implement. Let an observed time series, $v(t)$: $t = 1,..., N_D$, be the output of a differentiable dynamical system, f^t, on an m-dimensional manifold, M; i.e., f^t: $M \rightarrow M$, where f^t is the t^{th} iterate of f, is the nonlinear dynamics underlying the data and $v(t)$ is the measured time history of one of the coordinates for the orbit in M, $v(t) = S(x_t)$, $x_t \in M$. By Takens' theorem, when delays are introduced, an embedding of M into $\mathbb{R}^d$ typically results as long as the number of components, d, is made large enough. Smoothness properties of the dynamical system are now reproduced in the embedded image of M in $\mathbb{R}^d$. The delay vector time series.

$$x(t) = (v(t), v(t + \Delta),...,v(t + (d - 1)\Delta)), \qquad (2)$$

where Δ is the delay, and $t = 1,..., N = N_D - (d - 1)\Delta$, lives in that image. Its behavior carries the smoothness.

We denote the time-one map, i.e., f^1, by F and consider the following general quantity,

$$\phi = \phi(x) = \Psi(x, F^b(x),...F^{b(R-1)}(x)), \quad R > 1, \quad (3)$$

where F^b denotes the b^{th} iterate of F and where Ψ is some smooth function of its R vector arguments into $\mathbb{R}^d$. $\phi(x)$ is a vector field in

IR^d; i.e., the vector, ϕ, is (merely) a function of the vector, x. We take $b = 1$ here for simplicity. A simple form for $\phi(x)$ is

$$\phi(x) = \sum_{r=0}^{R-1} c_r F^r(x), \qquad R > 1. \tag{4}$$

F may be an arbitrarily sampled flow, or a map; $F^0(x(t)) = x(t)$, $F^1(x(t)) = x(t+1)$, etc. The c_r are at our disposal, and for simplicity, we take them to be constants, independent of x.

Directional (unit vector) fields for $\phi(x)$ for orbits of dynamical systems are smooth and depend on the choice of the c_r. To estimate such fields, we partition the phase space by a uniform grid. We call the j^{th} mesh cell of points, comprising the x_i, $i = 1,\ldots,n_j$, box-j, and we compute the average of the directional elements, $\hat{x} = \phi(x) \,||\, \phi(x) \,||^{-1}$, over box-$j$,

$$Y_j = n_j^{-1} \sum_{i=1}^{nj} \hat{x}(x_i). \tag{5}$$

For illustration, we compute the vectors, Y_j, for Lorenz system and Hénon map ($x' = 1 - 1.4x^2 + y$, $y' = 0.3x$) data and plot them (see Figure 3). The vector time series were computed from the delay coordinate construction on the x-coordinate for each case. We note the choice $\{c_r\} = \{-1, 1\}$ produces a directional field whose "arrows" point to the position of the next iterate. For finely sampled flows, this vector field approximates the flow line tangent vector field.

$||Y_j|| = 1$ if all unit vectors $\hat{x}$ are parallel in box-j. This should sensibly be the case for the smooth vector fields produced by most dynamical systems. If the time series is generated from a random system, however, instead of varying smoothly, the directions of the Y_j formally realized under the DCC will fluctuate irregularly. Moreover, the corresponding individual $\hat{x}(t_i)$ are almost surely not parallel in a given box-j, and $||Y_j|| \ll 1$ typically results.

A measurement on a phase portrait that is independent of the choice of vector field when the time series is deterministic and which captures these effects can be formed from a global average of the lengths of the local means of the directional

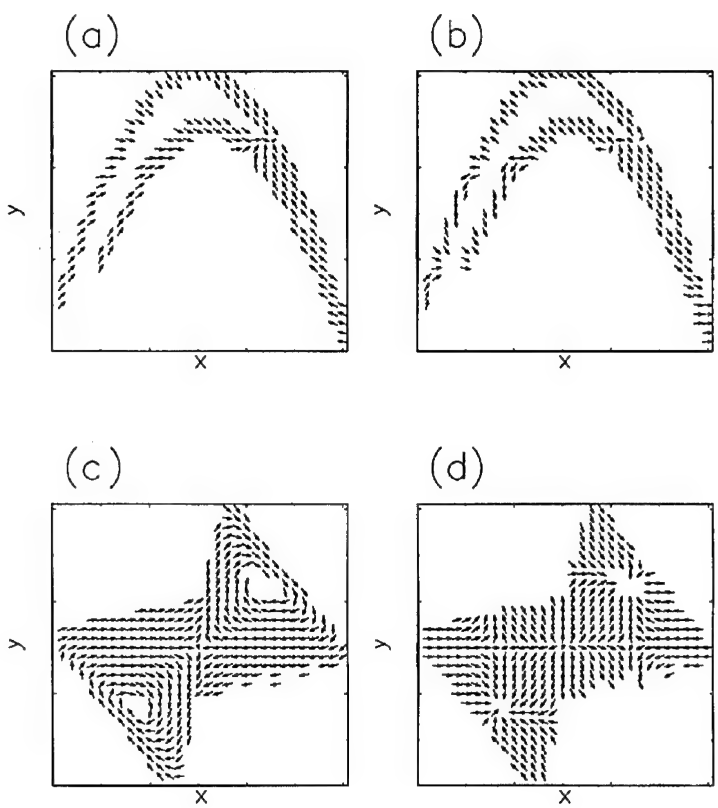


Figure 3. *Directional element fields for $d = 2$ for Hénon map (delay $\Delta = 1$) and Lorenz system ($\Delta = 4$). Data length $N_D = 20{,}000$, grid size $n^2 = 30 \times 30$. (a) Hénon: $\{c_r\} = \{-1, 1\}$. (b) Hénon: $\{c_r\} = \{2, -5, 3\}$. (c) Lorenz: $\{c_r\} = \{-1, 1\}$. (d) Lorenz: $\{c_r\} = \{2, -5, 3\}$.*

elements based on Equation (5); e.g., proceeding in the spirit of References 14 and 15,

$$W = N^{-1} \sum_j n_j \|Y_j\|^2. \qquad (6)$$

Practically any function of the $\|Y_j\|$ can serve; Equation (6) is just a weighted mean square.

Properties of *W*

For smooth data, $\|Y_j\| = 1$ for box-j sufficiently small, and $W = 1$ should result. In fact, owing to finite numerics, W is often a lot less than one. In particular, W depends on embedding parameters; for fixed d, $W = W(\Delta)$. W also depends on the choice of vector field ϕ. And the "natural" choice $\{c_r\} = \{-1,1\}$, implicit in the methods of References 14 through 16, does not necessarily produce the most deterministic looking $W(\Delta)$ (see Figure 4).

We observe that corresponding numerical data for the Λ-statistic of References 14 and 15 lie beneath the lower curves. The poorer performance of the $\{-1, 1\}$ vector field was not an isolated example but, in fact, was common. Computation parameters for the study in Figure 4 were $d = 3$ and $N_D = 20{,}000$ and the grid size, set by maximum range of the data, was $n^3 = 40 \times 40 \times 40$. We use these values henceforth.

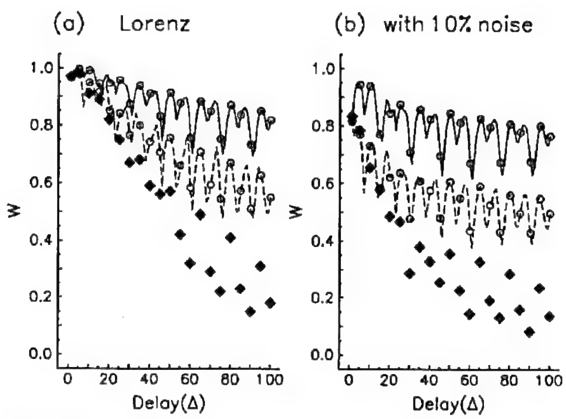


Figure 4. *Vector field dependence of computed values of W. $\{c_r\}$ =: {1, 2, 3, -4, -2} (solid curve), {-1, 1} (dashed curve). Also shown: comparison with Λ-statistic of References 14 and 15 (solid diamonds).*

W_M, a Stable Index of Smoothness

Since $W = 1$ is supposed to hold in the smooth case for any $\{c_r\}$, the choice of vector field is arbitrary. This gives us a tool to deal with the finite numerics problem just noted, for now we can exploit the very wide range of options available from Equation (4).

We list ten vector fields in Table 1; for convenience we have chosen $\Sigma_{r=0}^{R-1} c_r = 0$. We compute $W(\Delta)$ for each vector field; and for each delay, Δ, we further identify both maximum and minimum values of W; viz., $W_M(\Delta)$ and $W_m(\Delta)$, over the ten choices. The results are shown in Figure 5. The range of Δ for the Lorenz data, up to $\Delta = 3000$, corresponds to about 180 cycles.

Although the differences between values of $W_M(\Delta)$ and $W_m(\Delta)$ are systematic and large, as Δ rises, the upper envelopes of the $W(\Delta)$ plots descend to well-defined constant values. We denote these envelope values by W_M for the maximum and W_m for the minimum.

In the examples given in Figure 5, the values of W_M are close to one for both Lorenz ($W_M = 0.92$) and Hénon ($W_M > 0.99$) time series. Since there are a number of factors that may contribute to the finite numerics problem, such as the choices of vector field, grid size, embedding dimension d, input time series length, and sampling rate; we feel that $W_M > 0.9$ is a strict requirement. Accordingly, if $0.9 < W_M < 1$, we write $W_M \simeq 1$ to signal our acceptance that the test $W_M = 1$ is met.

Table 1. *Coefficients c_r for Ten Vector Fields ϕ_n, n = 1, 2,...,10, Used in Computations of W(Δ)*

	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}
c_0	-1.0	-3.0	2.0	4.7	-2.0	3.5	-3.4	1.0	0.9	3.0
c_1	1.0	4.0	-5.0	-3.0	3.0	-2.7	-0.5	2.0	0.8	-2.0
c_2	0.0	-1.0	3.0	-1.7	-4.0	-1.4	-0.1	3.0	-3.5	0.0
c_3	0.0	0.0	0.0	0.0	3.0	0.6	4.0	-4.0	4.0	2.0
c_4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-2.0	-2.2	-3.0

Indeed standard chaotic datasets sometimes fail to measure up to this requirement. The Ikeda map is one example of this, where we found $W_M = 0.87$, which is high, but not high enough for our requirement. We found $W_M = 0.89$ when we raised the embedding dimension for the computation to $d = 4$; so $W_M \simeq 1$ fails in both cases. Our Ikeda map[17] time series was x(n), where

$$z(n) = x(n) + iy(n),\ z(n+1) = 1.0 + 0.9z(n) \exp\left[0.4i - \frac{6.0i}{(1+|z(n)^2|)}\right].$$

Another example where $W_M \simeq 1$ fails is a relatively low-noise laboratory experimental time series that is known to be chaotic. The data represent the horizontal displacement of the base of a magnetostrictive ribbon. These data were taken in the first of the series of three experiments performed by the collaboration between the Nonlinear Dynamics Group and the Magnetism Group at NSWCDD White Oak (see Reference 18). Our measurement on these data gave $W_M = 0.85$.

Evidently, these results do not necessarily mean that $W_M \simeq 1$ cannot be achieved for the dataset. For example, some wider choice of vector fields might succeed, and a simple thing to do would be to enlarge the set used for the computation.

Nonetheless, it would seem that, unless we ease our choice of determinism tolerance, some chaotic datasets may, unfortunately and unjustly, be numerically separated from the smooth class. We address this problem in the next subsection by one method, and in the last section by a more radical and forceful one.

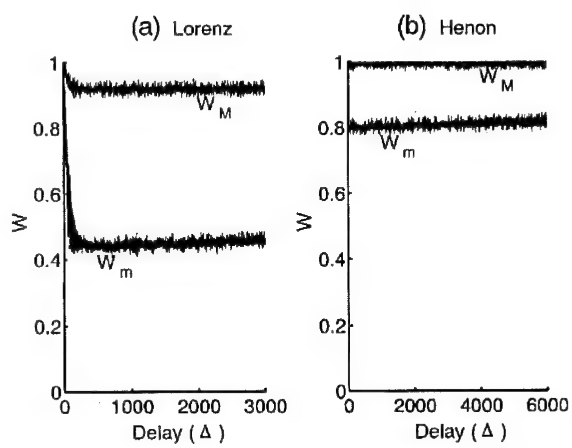


Figure 5. *Determinism case studies for (a) Lorenz and (b) Hénon time series. Computed W values shown are maxima ($W_M(\Delta)$) and minima ($W_m(\Delta)$) for each delay over the ten arbitrarily chosen vector fields in Table 1.*

Comparison Test — Making Use of W_m

We would like to accommodate high values of W_M that might signal the presence of determinism, but fail to meet the strict requirement, $W_M \simeq 1$. At the same time we also want to avoid relaxing our determinism tolerance standard. Accordingly, we have devised a fall-back position.

If $0.7 < W_M < 0.9$, we write $W_M \sim 1$, and consider how we might proceed in such cases. We adopt a method of statistical hypothesis testing in common use now in the nonlinear dynamics community. A null hypothesis, H_0, that the state of the world represented in the dataset belongs to a particular statistical class, is placed in opposition to an alternative hypothesis, H_1, which is a statistical complement of the null. Our fall-back method is a simple comparison test that can test the given data against selected classes of random processes. These classes, variously, then form a collection of nulls.

Following the implementation of statistical bootstrap from Reference 19, we make use of surrogate datasets to specify H_0 operationally. We use again the arbitrariness in the choice of vector field ϕ. As noted already in Reference 19, the surrogate class may be specified in any of a number of ways. For the examples below, we generated surrogate data from the Fourier transform (periodogram) of the given scalar time-series data by randomizing the phases and transforming back (Algorithm I in Reference 19). We did not introduce other nulls, although we stress the advisability of so doing. Using the vector fields in Table 1, we computed both W_M and W_m for the surrogate data and for the given data.

Organizing results as shown in Figure 6 for the Ikeda map, we easily distinguish the given data from the surrogate. For, suppose in Figure 6 the Ikeda map time series had been some other realization of the surrogate. Panels (a) and (b) would look exactly alike. This was actually the case for ambient ocean acoustic sonobuoy data (see Figure 7). Thus, using this method, the acoustic data plots show little evidence of smoothness and therefore determinism, while

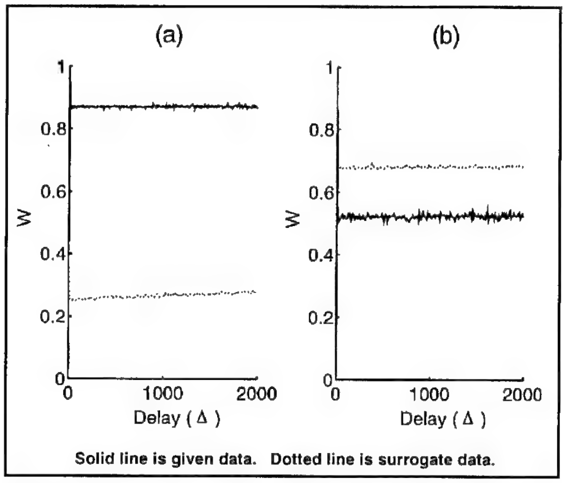


Figure 6. *$W_M(\Delta)$ and $W_m(\Delta)$ plots for comparison test analysis of Ikeda map data. For each Δ, W equals:*
(a) maximum for data, minimum for surrogate;
(b) minimum for data, maximum for surrogate.

the Ikeda map data show strong evidence of determinism.

When we applied this test to the ribbon dataset it also was clearly distinguishable from the surrogate; so, here again, the evidence for determinism is strong.[13]

Not Random at All?

From this point of view, we regard $W_M \sim 1$ as providing evidence for determinism if it can be

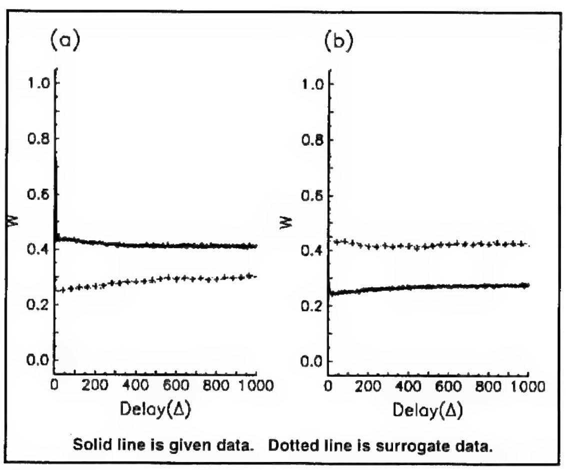


Figure 7. *$W_M(\Delta)$ and $W_m(\Delta)$ plots for comparison test analysis of ambient ocean acoustic data. For each Δ, W equals:*
(a) maximum for data, minimum for surrogate;
(b) minimum for data, maximum for surrogate.

supported by a negative result from the comparison test—that is, if the given dataset *can* be distinguished from one belonging to a random surrogate class.

We can strengthen (or refute!) our basis for a determinism call in the $W_M \sim 1$ case by implementing it repeatedly against a variety of surrogate classes. We agree with the emphasis in Reference 19, that this should be an important part of a serious protocol of "chaos hunting." In this way, we can build the case for concluding that a dataset cannot be distinguished from the smooth class.

Assuming we have done this, the resulting conclusion is still a little weaker than that issuing from $W_M \simeq 1$; for the latter means that the dataset cannot be distinguished from the smooth class at all. That is, it is consistent with determinism in the sense of being "not random at all." There is no need for surrogate comparisons.

We are still not done, however, for our dataset might have had enough noise to mask the presence of determinism. This innocent sounding statement leads to a radical departure.

Chaotic Noise Reduction—Widening the Net

"The use of noise-reduction methods is strongly recommended in any analysis of chaotic time-series data."[20] This is an integral part of our approach also, although now for radical new reasons as well as the salutary reasons of Reference 20.

We wish to have it that a conclusion we might frame for a given dataset will be grounded in standards that are unforgiving. In practice; i.e., without being over-restrictive, if we are to cut out false alarms, such standards should be as unforgiving as we can manage. That is why we have set the tight determinism tolerance requirements described in the previous section.

But we have found in controlled numerical studies that even small amounts of noisy contamination of a chaotic time series can quickly lead to $W_M < 1$. Thus, even if we have ruled clearly against determinism in some example, or have only been able to secure $W_M \sim 1$, the time series still might possess a deterministic part. That is, we might suppose that after a successful chaotic noise reduction, the output time series might no longer be distinguishable from the

smooth class, and we would have then a deterministic call. We'd like to catch these in our net, too.

This is actually a radical program, because we will be willing to say we have a chaos call on the basis of a time series derived from the given dataset, rather than just the initial dataset itself. The basis for isolation of the chaotic part will depend on the mathematics of the chaotic noise-reduction algorithm used. We also have the makings of an exactly similar kind of separation using a novel and different method—one based on predictability analysis,[21] but which we cannot discuss here. (The question to which we have been able to give precise formulation in Reference 21 is, "What is the sense in which one can say that a time series has a predictable component?" In principle this can be answered, and the answer suggests a way of isolating that component. This research is giving rise to interesting and valuable new wrinkles in time series analysis.)

In other words, integration of the use of chaotic noise reduction into our approach provides a way for us to hold more strictly to our tolerances against deterministic calls (fewer false alarms) and, at the same time, give the matter our best shot (more detections).

In the deterministic case, of course, the added benefit is that estimates of dynamical quantities like system dimension (number of active degrees of freedom), embedding dimension, various fractal dimensions, Lyapounov exponents, and the like, ought to be more reliable and accurate; i.e., when we use less noisy, cleaned-up versions of the data to perform them.

Effects of Noise on Phase Portraits

We now consider time series that are both chaotic and noisy. We assume the noisy part is some form of randomness. We denote the *given* dataset now by $v(t)$, $t = 1,2,\ldots,N_D$; without loss of generality, we may write

$$v(t) = V(t) + \varepsilon\eta(t),\ t = 1,\ldots,N_D, \qquad (7)$$

where $V(t)$ here represents the underlying deterministic part. We have written the random part of the time series as $\varepsilon\eta(t)$, where $\eta(t)$ is a zero mean process having unit variance.

Like typical forms of noise, chaotically generated time series have infinite bandwidth. Consequently, conventional band-passing methods for noise reduction can lead to undesirable effects, which noise reduction methods based on dynamical systems theory can avoid. See References 20 and 22 for recent surveys.

When the time series, $v(t)$, represents a (noise-free) dynamical system, the tip of the delay vector, $x(t)$, traces out a geometrical object, the embedded image of the attractor in $\mathbb{R}^d$. For a chaotic time series, this object is a fractal, but the flow lines, unstable manifolds, and dynamics all are smooth. Note also that the noise-free attractor image is contained in, and samples, the m-dimensional embedded image of the original true space for the dynamics, *viz.* M. But when the data are noisy, as in Equation (7), the orbit points in $\mathbb{R}^d$ fluctuate out of this embedded version of M, resulting in a "fuzzy" and not-so-smooth image of the underlying geometrical object. (Precisely, they fluctuate out of the embedded image, $M' \subseteq \mathbb{R}^d$, of M.)

One result of a successful application of chaotic noise reduction to a dataset will be to recover an underlying smoothness that would be there were it not for the noise.

Chaotic Noise Reduction

Evidently, when a raw dataset is given, we do not have the decomposition in Equation (7) available. For a noise reduction method to be meaningful (and successful!), it must correctly exploit some aspect of dynamical systems theory to provide a basis of identification of a deterministic part.

Reviews of a variety of different noise-reduction methods that exist are in References 20 and 22. These differ from one another according to which aspect is exploited. Typically, these proceed through an identifiable sequence of four steps.

Step 1. Embed; i.e., replace the given scalar dataset, $v(t)$, by a data-state vector, such as $x(t)$ in Equation (2). This provides a phase portrait in $\mathbb{R}^d$ for the Step 2 manipulations of the noise-reduction process.

Step 2. Adjust the positions of the points of the phase portrait. This is what is

normally regarded as the key noise reduction step, with the vector time series, $x(t)$, replaced by $\hat{x}(t)$.

Step 3. Disembed; i.e., replace the altered data-state vector time series, $\hat{x}(t)$, by a new scalar dataset, $\hat{v}(t)$. This step, the necessity of which was first realized in Reference 23, can also remove further noise.

Step 4. Iterate the foregoing, inputting $\hat{v}(t)$ in Step 1.

Sometimes the order of these steps is shuffled. For instance, in the trajectory adjustment step (Step 2) Kostelich and Yorke[24] iterate before disembedding. In their method, one can do it either way in principle, although that is not always the case.

The method used in Step 3 is quite important. Sometimes Step 3 is avoided entirely, with resulting loss in performance. The issue is this: the altered data-state vector time series, $\hat{x}(t)$, is almost surely not itself a DCC, but the right answer, namely the DCC from $V(t)$, surely is. To disembed, we construct the scalar time series, $\hat{v}(t)$, having the following property: namely, that the vector time series given by its DCC is as close as possible to the altered vector time series, $\hat{x}(t)$.

This gives our first guess at the noise-free scalar dataset, $V(t)$. For t-values not too close to endpoints of the time series, a simple least-squares criterion gives

$$\hat{v}(t) = d^{-1}\sum_{j=1}^{d} \hat{x}_j(t - (j - 1)\Delta), \qquad (8)$$

where $\hat{x}_j$ denotes the j^{th} component of the vector, $\hat{x}$. Similar expressions result for end-point range t-values. We note the smoothing effect of the averaging that occurs in Equation (8). For details of the full noise-reduction process using the local geometric projection (LGP) method, see Reference 23.

Of course, different disembedding norms besides the least squares, (L^2), are possible. The once popular choice in the community, $\hat{v}(t) = \hat{x}_1$, namely, that of the first component of $\hat{x}$, is a poor relation of the L^p disembedding with $0 < p < 1$. This also gives a single component, but usually not the first component!

We remark that iteration is an essential part of chaotic noise reduction since all the methods so far proposed in the research literature are based on successive approximations. Otherwise, improvements are very small. This circumstance has significant and useful consequences involving attractor stability and instability under iteration,[25] and quantitative measures of such effects.[26] This creates opportunities for chaotic time series analysis, which we cannot go into here.
(Among these is a method under development at NSWCDD for control of chaotic noise reduction.)

Smoothness after Noise Reduction

Even low levels of noise can interfere with the determinism test.

We applied the LGP algorithm to the ribbon data, where the noise is dynamical with $SNR_i \sim 35dB$, estimated by the method of Reference 26. We computed W_M using the vector fields in Table 1. This time, we got $W_M = 0.91$, gaining now $W_M \simeq 1$ for the noise-reduced version. For comparison, we recall the value, $W_M = 0.85$, found for the raw dataset earlier.

So the (noise-reduced) ribbon data now are consistent with "not random at all."

As a control we subjected the Ikeda map dataset to noise reduction. We found that W_M was unchanged, giving again only $W_M \sim 1$. Noise reduction had no discernible effect on the ambient acoustic noise time series of the last section, and W_M again obeyed a clear $W_M < 1$.

Another example of an experimental dataset we examined is a time series ($N_D = 12{,}000$) consisting of intensity measurements of a laser in a chaotic state. The data were reported and analyzed in Reference 27. We found $W_M = 0.83$, so we can say only $W_M \sim 1$. We applied the noise-reduction algorithm to the laser data. W_M only increased slightly and still lay in the range $W_M \sim 1$. So here we were unable to conclude consistency with "not random at all."

A couple of other examples are interbreath interval data for sheep fetuses (no chaos) and time-dependent strain data from a free-running spindle (deterministic, but only periodic; and no underlying chaos).

Summary and Discussion

We summarize our philosophy of approach to a dataset in a flow chart (Figure 8). When we get to the bottom of the chart, whether the end result has been conclusive or inconclusive, it is vital still to go back and retest noise-reduced versions of the data.

Suppose our analysis has brought us to a conclusion that our data, or a noise-reduced version of our data, are indistinguishable from the smooth class and that we do want to go further with the analysis. There are many things we *should* then do; there are many excellent ideas in the literature now. Reference 28 contains several of these in conjunction.

No method can be better than its algorithms—in the present case, the smoothness detector and noise reduction algorithm. What we have done in this article is to advocate a fairly stern procedure, but it is only a procedure for getting started, for finding a basis for a yes/no decision in the analysis of a dataset.

We think this should be the first step always taken.

Acknowledgments

This work was supported by the Office of Naval Research, Physical Sciences Division (M. F. Shlesinger) and by the NSWCDD Independent Research Program.

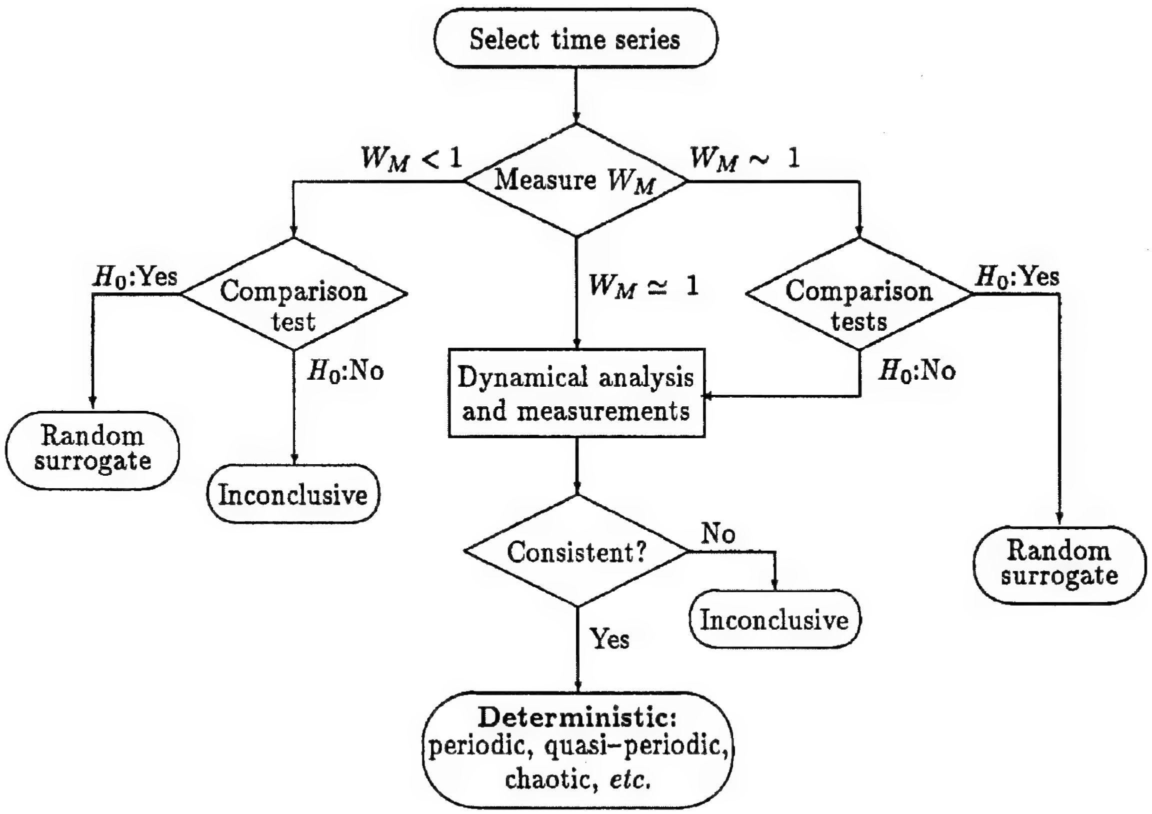
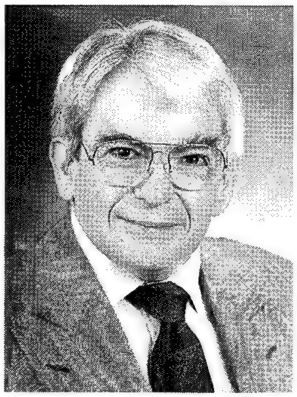


Figure 8. *Flow chart for time series analysis to discriminate against smoothness.*

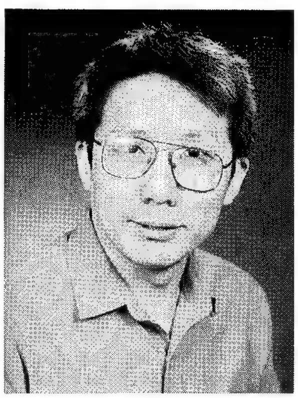
References

1. Newton, I., *Philosphiae Naturalis Principia Mathematica*, E. Halley, 1686.
2. Poincaré, H., *Les Methodes Nouvelles de la Mecanique Celeste*, Gauthier-Villars, Paris, 1892 (*t*.1), 1893 (*t*.2), and 1899 (*t*.3). (For a recent translation see H. Poincaré, *New Methods of Celestial Mechanics*, Vols. 1-3, Daniel L. Goroff, ed., AIP, 1993.)
3. Birkhoff, G. D., *Dynamical Systems*, American Mathematical Society Colloquium Publications, Vol. IX, 1927.
4. Anosov, D. V., Proc. Steklov Inst. Math. 90, 1 (trans. from Trudy Mat. Inst. Steklov).
5. Smale, S., *Bull. Amer. Math. Soc.* 73, 747, 1967.
6. Lorenz, E. N., J. *Atm. Sci.* 20, 130, 1963.
7. Mandelbrot, B. B., *The Fractal Geometry of Nature*, W. H. Freeman & Co., New York, 1982.
8. Ditto, W. L.; Spano, M. L.; Savage, H. T.; Rauseo, S. N.; Heagy, J.; and Ott, E., *Phys. Rev. Letters*, 65, 533, 1990.
9. Spano, M. L. and Ditto, W. L., "*Taming Chaos Experimentally: a Primer*," in *Procs. 1st Experimental Chaos Conference*, 1-3 Oct 1991, Arlington, VA, Sandeep Vohra, Mark Spano, Michael Shlesinger, Lou Precora, and William Ditto, eds., World Scientific, Singapore, 1992, p. 107.
10. Packard, N. H.; Crutchfield, J. P.; Farmer J. D.; and Shaw, R. S., *Phys. Rev. Letters*, 45, 712, 1980.
11. Takens, F., "Detecting Strange Attractors in Turbulence," *Dynamical Systems and Turbulence, Warwick, 1980*, Lecture Notes in Mathematics, No. 898, D. A. Rand and L. S. Young, eds., Springer-Verlag, Berlin 1981, p. 366.
12. Cawley, R.; Hsu, G.-H.; and Salvino, L. W., "Method to Discriminate Against Determinism in Time Series Data," *Predictability of Complex Dynamical Systems*, James B. Kadtke and Yurii Alexandrovich Kravtsov, eds., Springer-Verlag, Berlin (to appear).
13. Salvino, L. W. and Cawley, R., *Phys. Rev. Letters* 73, 1091, 1994.
14. Kaplan, D. T. and Glass, L., *Phys. Rev. Letters* 68, 427, 1992.
15. Kaplan, D. T. and Glass, L., *Physica D*, 64, 431.
16. Wayland, R.; Bromley, D.; Pickett, D.; and Passamante, A., *Phys. Rev. Letters* 70, 580, 1993.
17. Hammel, S. M.; Jones, C. K. R. T.; and Moloney, J. V., *J. Opt. Soc. America*, B 2, 552, 1985.
18. Ditto, W. L.; Rauseo, S. N.; Cawley, R.; Grebogi, C.; Hsu, G.-H.; Kostelich, E.; Ott, E.; Savage, H. T.; Segnan, R.; Spano, M. L.; and Yorke, J. A., *Phys. Rev. Letters* 63, 923, 1989.
19. Theiler, J.; Eubank, S.; Longtin, A.; Galdrikian, B.; and Farmer, J. D., *Physica D* 58, 77,1992.
20. Kostelich, E. J. and Schreiber, *T., Phys. Rev. E* 48, 1752, 1993.
21. Salvino, L. W.; Cawley, R.; Grebogi, C.; and Yorke, J. A., "Predictability in Time Series," *Phys. Lett. A* 209 (1995) pp. 327-332.
22. Grassberger, P.; Hegger, R.; Kantz, H.; Schaffrath, C.; and Schreiber, T., *Chaos* 2, 127, 1993.
23. Cawley R. and Hsu, G.-H., *Phys. Rev. A* 46, 3057, 1992.
24. Kostelich, E. and Yorke, J. A., *Phys. Rev. A* 38, 1649, 1988.
25. Cawley R. and Hsu, G.-H., "Another New Chaotic Noise Reduction Algorithm," *Proceedings of the First Experimental Chaos Conference*, 1-3 Oct 1991, Arlington, VA, Sandeep Vohra, Mark Spano, Michael Shlesinger, Lou Pecora and William Ditto, eds. World Scientific Press, Singapore, 1992, p. 38.
26. Cawley R.; Hsu, G.-H.; and Salvino, L. W., "Detection and Diagnosis of Dynamics in Time Series Data: Theory of Noise Reduction," *The Chaos Paradigm: Developments and Applications in Engineering Science*, AIP Conference Proceedings Vol. 296, American Institute of Physics, New York, 1994, p. 182.
27. Hübner, U.; Weiss, C.-O.; Abraham, N. B.; and Tang, D., "Lorenz-like Chaos in NH_3-FIR Lasers," *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend and N. A. Gershenfeld, eds., Addison-Wesley, Reading, Mass., 1993, p. 73.
28. Tufillaro, N. B.; Wyckoff, P.; Brown, R.; Schreiber, T.; and Molteno, T., *Phys. Rev. E* 50, 164, 1995.

The Authors



ROBERT CAWLEY received a B.S. in physics from the Massachusetts Institute of Technology in 1958, and an M.S. and Ph.D. in physics, with minor in astronomy, from the University of Illinois, Urbana, in 1960 and 1965, respectively. He taught physics at Clarkson University from 1965 to 1967, and came to NSWCDD/White Oak in 1967. In 1983 he started the NSWC Navy Dynamics Institute Program at White Oak with the help of ONR funding. He is currently a Principal Research Scientist at NSWCDD and Senior Staff Scientist to the Information Science and Systems Branch. He received a Navy Meritorious Civilian Service Award in 1991 for his nonlinear dynamics work at NSWC, and a lifetime achievement award by NUWC/DIVNPT in 1995 for contributions and support of nonlinear dynamics in full spectrum processing. His current scientific interest is applications of nonlinear dynamics and fractals to physical processes and to experimental time series analysis. He is currently engaged in the pursuit of technology applications to a range of manufacturing processes and to power distribution. Dr. Cawley is a member of the American Physical Society, the American Mathematical Society, and the Society for Industrial and Applied Mathematics.



GUAN-HSONG HSU received his B.S. in mathematics from the National Taiwan University in Taipei, Taiwan, Republic of China, in 1978, and his M.S. and Ph.D. in applied mathematics from Courant Institute, New York University, in 1983 and 1986, respectively. He was a postdoctoral associate shared between the University of Maryland and NSWCDD from 1986 to 1989. He taught mathematics at the University of Missouri, Columbia, from 1989 to 1992. He was an ONR Postdoctoral Fellow at NSWCDD from 1992 to 1995. He is currently a Research Scientist in the Physics Department at American University. His scientific interest is in various areas of scientific computations: numerical methods in chaotic dynamics and their applications, especially to data analysis; and computational fluid dynamics, especially numerical weather models and computational geometry. Dr. Hsu is a member of the American Mathematical Society and the Society for Industrial and Applied Mathematics.

LIMING W. SALVINO received her B. S. in physics from the Northern Jiao-Tong University, Beijing, China, in 1982, and her M.S. and Ph.D. in physics from The American University, Washington, D.C., in 1986 and 1989, respectively. After graduation, she was a research consultant at the American University, Department of Physics from 1989 to 1990. In 1990, she became a research associate and postdoctoral fellow at the University of Maryland and NSWCDD, and has continued at NSWCDD through the ONR Postdoctoral Fellowship program since 1993. Her initial research efforts were in statistical mechanics of phase transitions, numerical analysis, computer modeling, and molecular dynamics simulations of fluids and fluid flows. She is currently interested in applications of nonlinear dynamics and chaos to data analysis, especially time-series modeling and prediction. Dr. Salvino is a member of the American Physical Society.

The NSWCDD Mathematics and Statistics Libraries

John R. Crigler

This article traces the creation and evolution of two computer libraries developed at the Naval Surface Warfare Center, Dahlgren Division (NSWCDD). Initially developed in the late 1970s, the NSWCDD Library of Mathematics Subroutines (NSWCLIB) is a collection of general purpose mathematical software. Developed in the late 1980s, the NSWCDD Library of Statistical Programs and Subroutines (STATLIB) is a specialized collection of software in the areas of probability and statistics. The impetus for development, the usage, and the impact that each of these libraries has had on the scientific community, both within and outside of NSWCDD, is discussed.

Introduction

Ever since the advent of high-speed digital computers, there has been an increasing demand for highly reliable scientific and engineering routines to support a wide variety of research and analysis efforts. In response to this need at NSWCDD, two computer libraries emerged. Both were written in the Fortran computer language. The NSWCLIB was initially developed in the late 1970s to provide reliable, general purpose mathematical software to the NSWCDD scientific community. Within NSWCDD, this library is known as MATHLIB. Since its first release in 1978, subsequent editions have been greatly expanded, and NSWCDD has distributed the library to numerous sites both in the U.S. and abroad. The library has received high praise for its breadth of coverage, the exceptional reliability of its code, and its transportability. A code is considered to be reliable if it does not fail and if it achieves its specified accuracy. The transportability of a library of computer routines refers to the extent to which the library can be installed on different computers.

In an independent effort, a second library, STATLIB, was developed in the late 1980s to provide a specialized set of programs and subroutines in the areas of probability and statistics to Dahlgren Division scientists and engineers. The STATLIB was developed as a complement to commercial packages available at NSWCDD at the time.

The purpose of this article is to trace the development and evolution of NSWCLIB and STATLIB and to provide a sense of the impact that each has had on the scientific community, both within and outside of NSWCDD.

NSWCLIB

Development of NSWCLIB

The development of NSWCLIB began in 1976 at the urging of Ralph A. Niemann, then Head of the Warfare Analysis Department. At that time, almost no reliable general purpose numerical mathematics software was available. No software engineering standards had been established, and the existing codes at NSWCDD were, in general, quite deficient. Alfred H. Morris, Jr., a senior research

mathematician in the department, reports that he was asked to do whatever he could to assemble a reliable collection of Fortran codes in a form that could be made available to the entire NSWCDD scientific community.[1] Dr. Allen V. Hershey, a senior research scientist in the Warfare Analysis Department, was assigned by Niemann to assist Morris whenever time permitted. At that time, it was not clear what software standards should be imposed or how the project could be organized in a cost effective manner.

The library building project was considered to be high risk for a number of reasons. In Reference 1, Morris remarks that some of the problems encountered in library development included organization, location of required manpower to properly support the development, and the procurement or development of reliable code. Developing a highly reliable routine can be quite involved. To begin with, the numerical solution to a problem frequently requires using several different algorithms so that the routine can accommodate a wide variety of values of the input parameters. Further, one or more of these algorithms may possess regions of numerical instability that require special treatment by the analyst. Failure to find an acceptable coding solution to just one of these problem areas was deemed by Morris to be a sufficient reason for rejecting a routine's inclusion in NSWCLIB.

The first edition of NSWCLIB was published by Morris in June 1978 as an NSWCDD technical report.[2] That edition contained 84 functions and subroutines organized in 11 distinct sections. Since that first release, the library has grown enormously. To date, there have been seven editions.[2-8] The latest, released in January 1993, contains 1062 functions and subroutines. Of these, 576 are intended for general use and are documented in the report in 26 distinct sections. The remaining ones are classified as support routines and are normally of little interest to most users. These supportive routines perform pieces of the required computations, and their codes are very highly specialized. They are usually replaced as improved versions become available, without affecting the use of the documented routines. Approximately 40 percent of the routines in the current version of the library were developed at NSWCDD. The remaining routines were obtained from government, commercial, and university research centers, both in the U.S. and abroad. Of those developed at NSWCDD, the vast majority were written by Morris. Other NSWCDD contributors include Dr. Allen V. Hershey, Dr. Armido R. DiDonato, Dr. Milton P. Jarnagin, Richard K. Hageman, Dr. Andrew H. Van Tuyl, Dr. John R. Crigler, and Richard Pasto. Morris states that two of these individuals, DiDonato and Van Tuyl, have contributed codes that are extensively used in many of the library routines.

Although the routines in NSWCLIB were initially intended for use on the CDC 6000-7000 series mainframe computers, library development focused heavily on its transportability. The 1993 edition of the library is used on a wide variety of computers, from the CRAY Y-MP supercomputer to IBM-compatible personal computers. The current documentation[8] contains an appendix explaining the process for installing the library on any computer. It should be emphasized that NSWCLIB contains no machine-dependent code. However, machine-dependent constants and precision-dependent algorithms cannot be avoided. For a code to be acceptable, Morris required that it conform to the 1977 American National Standards Institute (ANSI) Fortran standard. In addition, no proprietary or otherwise restricted codes are included in the library. Morris noted that usage restrictions can severely impair a library's value, both for theoretical purposes and for general use in applications. Consequently, NSWCDD policy has been to make the source code readily available to the scientific community. Factors that influenced adoption of this policy include:

- Mathematical expertise is widely scattered.
- Developing reliable codes is extremely difficult.
- Many existing libraries do not provide source code.

Since the main goal of NSWCLIB is to provide a service to as broad a user community as possible, its ease of use was of prime concern. In support of this goal, no input/output (I/O) statements were permitted in the library codes. Morris permitted error-detection code with certain restrictions. If error-detection code was included in a function, then the function was required to be assigned a special value. If error-detection code was included in a subroutine, then the subroutine call line was required to contain one or more parameters for

error reporting. These parameters enable the user to maintain total control over the sequence of events that follow.

Before a candidate routine was accepted for inclusion in the library, it was thoroughly examined and tested. Morris' acceptance criteria included reliability, transportability, efficiency, and ease of use. The limits of applicability of a candidate routine were also considered; routines were always subject to reexamination and the possibility of subsequent modification. Morris remarks that a substantial amount of research in the last decade has led to the development of new and improved algorithms designed to handle a variety of problems. Whenever these results affected current library codes, those codes may have been rendered obsolete. If a library routine became obsolete, Morris would consider eliminating it. With respect to the reliability of a candidate code, Morris' main concerns were the accuracy of the code, the stability and robustness of the algorithms being used, and the code's overall quality. The term robustness refers to the range of parameter values that the algorithm can accommodate. The more extensive the range of possible values, the more robust the algorithm is said to be.

The NSWCLIB contains both single- and double-precision routines. Single-precision routines are designed for single-precision floating arithmetics having 6 to 14 digits of accuracy, and double precision routines are designed for double-precision arithmetics that are accurate to 12 to 30 digits.

Contents of NSWCLIB

The 1993 edition of the library contains approximately 115,000 lines of code. The headings of the 26 sections in the library are displayed in Table 1. The treatment that each of these mathematical areas has received in the library varies considerably. Morris specialized in matrix and special function theory, and the broad coverage of these areas in the library reflects his interest. The sections on elementary operations, vectors, curve fitting, and continuous random number generation also contain a substantial number of routines. The library features excellent spline-under-tension curve-fitting routines. On the other hand, there are no nonlinear programming routines in the library, and only one routine for solving partial differential equations is included. Morris noted that this variability of coverage is characteristic of all current general purpose libraries, not just NSWCLIB.

Usage of NSWCLIB

As NSWCLIB has grown over its 18 years of development, its usage within NSWCDD has steadily increased. Not only is it being used on CRAY mainframe computers, but on a variety of other machines as well. The list includes, but is not limited to, VAX and micro VAX computers, SUN workstations, and IBM-compatible personal computers. The library is used by many departments in support of a variety of tasks, both classified and unclassified. No hard data is available regarding the frequency with which the

Table 1. *Section Headings in NSWCLIB*

Elementary Operations	**Optimization**
Geometry	**Transforms**
Special Functions	**Approximation of Functions**
Polynomials	**Curve Fitting**
Solutions of Nonlinear Equations	**Surface Fitting over Rectangular Grids**
Vectors	**Surface Fitting over Arbitrarily Positioned Data Points**
Matrices	**Manifold Fitting**
Large Dense Systems of Linear Equations	**Numerical Integration**
Banded Matrices	**Integral Equations**
Sparse Matrices	**Ordinary Differential Equations/Initial Value Problems**
Eigenvalues and Eigenvectors	**Partial Differential Equations**
l_1 Solution of Linear Equations	**Discrete Random Number Generation**
Least-Squares Solution of Linear Equations	**Continuous Random Number Generation**

various sections of the library have been used at NSWCDD. However, user comments and questions regarding the usage of specific routines for project applications have provided a sense of the overall impact that the library has had and continues to have on research and development at NSWCDD. Some of the reported uses of the library routines are given in Table 2.

The content of Table 2 is by no means intended to be complete. It does suggest, however, the extent to which the library has been utilized and the impact that it has had on the NSWCDD scientific community. Those researchers and analysts who have availed themselves of the library as a resource tool have consistently given it high marks. Morris received recurring requests over the years to develop routines that are not yet in the library. One such request led to the development of a routine to compute the exponential of a real matrix. Such a facility had been badly needed for some time for use at NSWCDD in trajectory computations. Morris' development of this routine is particularly noteworthy, because this had been an unsolved computational problem for some 30 years. The theoretical breakthrough on this problem is attributed to Robert C. Ward of the Nuclear Division of the Union Carbide Corporation in Oak Ridge, Tennessee. Ward's solution,[9] published in a paper in 1977, avoided the problems of numerical instability that had plagued earlier proposed solutions. Morris coded the Ward algorithm and included it in NSWCLIB.

As further examples of requested codes, Morris cites nonlinear programming routines, more surface fitting routines, and additional optimization codes. He notes that many of these mathematical facilities have not been provided in the library because of the lack of the required expertise or manpower at NSWCDD to develop them. In addition, some requested routines have simply not been provided because no one has solved the computational issues involved, either at NSWCDD or elsewhere.

The NSWCLIB has become quite popular outside of NSWCDD in recent years. To date, over 500 copies of the library have been distributed to individuals and sites in the U.S. and abroad. Morris reports that some sites have distributed the library to other sites, and that copies of the library codes have also been distributed at conferences. Not much is known about the extent of this external distribution. Many of

Table 2. *Reported Usage of NSWCLIB Routines at NSWCDD*

Type of Routine	Area of Application
Least-squares solutions for systems of equations	Prediction of gravity anomalies from satellite altimetry data
Bessel functions	Computation of satellite drag Missile error analyses
Sorting lists	Trajectory computations
Optimization	Function development in trajectory modeling
Numerical integration	Satellite optimal control Trajectory equations of motion
Random number generation	Monte Carlo simulation
Matrix operations	Fire control computations
Ordinary differential equations	Trajectory generation
Eigenvalues/Eigenvectors	Control theory computations
Circular and elliptical coverage functions	Weapons accuracy studies

the subroutines in the library form a basis for a number of research and development results being published worldwide today.

In July 1992 a questionnaire was sent to 100 sites in the U.S. requesting opinions on the quality of the documentation and code, and comments on library utilization. Thirty sites responded; some of the applications reported that are of interest to the Navy are listed in Table 3.

The sites at which the library is used in the U.S. fall into three categories: (1) government R&D laboratories, (2) commercial R&D laboratories, and (3) universities. Some examples of sites in each category are given in Tables 4 through 6. Morris notes that the library is used by the military outside of NSWCDD on many projects. Some universities, such as Princeton, use the library primarily for research and development; while others, such as West Point, use it mainly for teaching purposes.

Outside of the U.S., the library has been most popular in Australia. Morris reports that the library has been distributed by NSWCDD to more than 40 sites in that country. These include primarily government R&D centers and universities. It is used by the Commonwealth Scientific and Industrial Research Organization (CSIRO), the Defence Science and Technology Organization (DSTO), and the Australian Defence Force Academy. The academy includes all branches of the military. CSIRO is Australia's main scientific research body. It is run by the government and employs approximately 5,500 people in 70 laboratories. DSTO performs the R&D function of the Australian Department of Defence.

General reaction to the library is that it is relatively easy to use and that its codes operate quite satisfactorily in a variety of computer environments. Morris remarks that many of the routines in the library have been converted

Table 3. *Usage of NSWCLIB Outside NSWCDD: Reported Naval Applications in the U.S.*

Site	Reported Applications
Catholic University	Acoustic wave motion
University of Wisconsin	Modeling of electromagnetic structures
Massachusetts Institute of Technology	Ballistic trajectory analysis
Institute for Management Science and Engineering	Logistics, risk analysis, military operations research
Software Integrity Corporation	Thermal analysis and engineering, CRT electron gun design
Webb Institute of Naval Architecture	Error analysis in shipbuilding, Monte Carlo simulation of ship ventures
Motorola	Characterization of RF scattering environments
National Oceanic and Atmospheric Administration	Water wave motion

Table 4. *Usage of NSWCLIB Outside NSWCDD: Government R&D Laboratories in the U.S.*

- Naval Ocean Systems Center
- Department of the Army - Coastal Engineering Research Center
- Strategic MissileTest Center - Hill Air Force Base
- Argonne National Laboratory
- Lawrence Livermore National Laboratory
- Los Alamos National Laboratory
- White Sands Missile Range
- NASA Ames Dryden Flight Research Facility
- NASA Center for Space Propulsion Engineering

Table 5. *Usage of NSWCLIB Outside NSWCDD: Commercial R&D Laboratories in the U.S.*

- Federal Electric Corporation -Western Space and Missile Center
- General Electric Company
- Westinghouse Defense and Electronics Center
- Calspan Corporation
- GTE Laboratories
- Pittsburg Supercomputing Center
- Minnesota Supercomputing Center
- Boeing Aerospace
- Pratt andWhitney Aircraft

Table 6. *Usage of NSWCLIB Outside NSWCDD: Universities in the U.S.*

- U.S. Military Academy - West Point
- Cornell University
- Princeton University
- Stanford University
- Harvard-Smithsonian Center for Astrophysics
- University of California at Berkeley
- Carnegie Mellon University
- University of Minnesota - Army Research Center

from Fortran to other programming languages, both at NSWCDD and elsewhere. However, the extent of the usage of such modified codes is unknown.

Relationship of NSWCLIB to Other Libraries

At present there are only a few transportable, general purpose mathematics libraries in existence that are highly regarded for their coverage and quality control. Examples on the same level as NSWCLIB include two commercial libraries—the International Mathematical and Statistical Libraries (IMSL) and the Numerical Algorithms Group (NAG) library. While it is true that these libraries do contain some duplicate facilities, they also tend to provide diverse capabilities.

Morris reports that during its 18 years of development, the NSWCLIB project has outlived most other noncommercial library building projects. These noncommercial libraries were developed by corporations and government research centers to provide reliable routines for in-house use. Some of these were terminated due to failure, while others, such as the Sandia Laboratories library, were terminated due to cost considerations. Some universities in the U.S. have released mathematical software packages, but they have usually been specialized. For example, a curve-fitting package devoted to B-splines was developed by Carl de Boor of the Mathematics Research Center at the University of Wisconsin. No U.S. universities have attempted to build comprehensive libraries. In light of this, Morris feels that the termination of comprehensive noncommercial mathematics library projects is indeed a sad state of affairs for the scientific community.

Most major general-purpose mathematics libraries existing today are commercial libraries, which must be leased. Morris remarks that libraries are normally used for two purposes: (1) production and (2) research and development. Object codes are of prime importance in the former; source codes are of prime importance in the latter. Most leased libraries either do not provide source codes or severely restrict the use of their source codes. Morris feels that this policy has always had an adverse impact on research and development in the U.S. and that it has contributed to a dramatic rise in research and development costs. He argues that source codes frequently contain the most recent theoretical and software engineering advances and that these codes are often the only source for such advances. This scarcity of available source code was a key factor in NSWCDD's decision to freely share the source code in the NSWCLIB.

Requests for copies of NSWCLIB from within NSWCDD should be directed to William J. Fairfax of the Strategic and Space Systems Department (540-653-7136). Requests for copies of the source code and documentation from outside of NSWCDD should be directed to the Defense Technical Information Center (800-225-3842 or 703-274-7633) if the request is from a government organization or a registered contractor. For all others, documentation can be obtained from the National Technical Information Service (800-553-6847 or 703-487-4650), and source code can be obtained by sending a self-addressed stamped floppy mailer (5.25-inch or 3.5-inch) to the following address:

COMMANDER
ATTN: CODE K11
NAVSURFWARCENDIV
17320 DAHLGREN ROAD
DAHLGREN VA 22448-5100

The technical point of contact for questions concerning usage of NSWCLIB routines is Dr. Armido R. DiDonato, a research associate in the Strategic and Space Systems Department at NSWCDD (540-653-8036).

Availability of NSWCLIB

NSWCDD is committed to ensuring that the 1993 edition of NSWCLIB continues to be made available to all interested users, both within and outside of NSWCDD.

Concluding Remarks on NSWCLIB

The NSWCLIB building project has been extremely successful. In view of its unique capabilities, many mathematics problems can now be handled routinely that would otherwise be exceedingly difficult or impossible to solve. Software standards have been established and new techniques such as sparse matrices and cubic splines have been introduced. Morris estimates that the 1993 edition of the library contains approximately $5 to $10 million worth of code! Yet he feels that this development cost has been only a small portion of its estimated research and development value. Its widespread impact on the entire NSWCDD scientific community implies that it must be considered to be one of the most cost effective projects ever undertaken at NSWCDD.

Morris' retirement from NSWCDD in July 1993 brought an end to full-time development of the NSWCLIB building project. However, NSWCDD has recently entered into a contractual agreement that will allow Morris to expand the library's capabilities on a part-time basis. As a result of this agreement, NSWCDD plans to support future releases of NSWCLIB as Morris provides significant upgrades. The library's exceptional quality guarantees that it will continue to have a far-reaching influence in the entire scientific community for many years to come.

STATLIB

Development of STATLIB

Statistical analysis has been employed at NSWCDD since the earliest days of ordnance testing. In response to the increased requirements for and complexity of statistical analysis, the Mathematical Statistics Branch was formed in the Warfare Analysis Department in 1963. At that time, statistical computation and analysis was conducted on mechanical desktop machines and computer programs written on an as-needed basis. Reliable commercial software as we know it today did not exist. As a result of this environment, the development of statistical software became a by-product of the branch almost from its very beginning. At the outset, very little actual programming was done within the branch. Programming requirements were formulated, and programming support was obtained from available sources within the department. This process continued well into the 1970s. At that time, the growth of the branch was sufficient to permit statistical software development to be conducted entirely from within. Consequently, many different programmers were involved in the development of the emerging software. With the exception of those programs that are documented in NSWCDD reports, the identities of most of the original programmers are unknown. As commercial statistical software became more available, more reliable, and easier to use, the need for developing new statistical software in-house was considerably reduced.

As the result of a reorganization in 1980, the Mathematical Statistics Branch became the Mathematical Statistics Staff in the Strategic Systems Department. In late 1984, NSWCDD's general-purpose mainframe computer, the CDC 6700, was in the process of being phased out and replaced by a newer model, the CDC CYBER 170-865/875. During that time, it became apparent that the entire collection of statistical routines developed over the years could not and should not be converted for use on the CDC 865/875. Some routines were designed to compute descriptive statistics and had been rendered obsolete by new commercially available software. Others were large and quite slow by the standards of that time. Consequently, these were deemed unworthy of the effort they would require in conversion and checkout. Application of such selection criteria resulted in the identification of some 30 programs marked for retention. It was also recognized that a few additional programs could be retained if they were structured in the form of subroutines for better utilization.

With the obsolescence of computer card files, the creation of an electronic computer library was deemed to be the most efficient means of storage that offered rapid access. The main impetus for establishing such a library was to improve the efficiency with which the Mathematical Statistics Staff performed its primary role of providing consulting and analysis services to NSWCDD

scientists and engineers. Although the members of the Mathematical Statistics Staff would undoubtedly be the prime users of the library, its establishment would make the software accessible to the entire NSWCDD community.

The development of the library, known as STATLIB, took place over approximately a four-year period from 1985 to 1989. Since the Mathematical Statistics Staff's primary mission was to provide consulting and analysis services to NSWCDD personnel, the development of STATLIB was strictly a part-time effort. Work was performed on the STATLIB project only when the group's regular workload was light enough to permit it. Three members of the group were primarily responsible for the development—Dr. Marlin A. Thomas, Gary W. Gemmill, and Dr. John R. Crigler. Thomas was Head of the Mathematical Statistics Staff at that time and should be credited with initiating the project. In addition, Ms. Elissa A. Zizzi converted the older programs in STATLIB from the CDC 6700 to the CDC 865/875, and Ms. J. Diane Bell set up the system library procedures. Gary M. Johnson, who joined the group in 1988, was also involved in the latter stages of testing of the STATLIB software.

With regard to the development of STATLIB programs, the procedure that was followed consisted of:

- Carefully reviewing the existing Fortran code, updating, where appropriate
- Correcting any discovered errors
- Designing a set of comprehensive test cases

In several programs, new capabilities and features were incorporated. With respect to STATLIB subroutines, the procedure involved the reconfiguration of old programs into subroutines and the programming of newly formulated subroutines. Comprehensive test cases were also constructed for all subroutines. Error-detection code was included in each subroutine, and the subroutine call line contained a single parameter for error reporting. The subroutines in STATLIB are all random-number generators. The decision to add new subroutines was based on the desire to offer the STATLIB user a fairly complete set of such generators for the common discrete and continuous probability distributions. I formulated and programmed the random-number generators in STATLIB. Both the programs and subroutines in STATLIB were extensively tested for correctness. While error-free software is a rare commodity, it is believed that STATLIB contains very few errors. This is due to the long history of use and the recent testing of the old software, as well as the extreme care taken in the development of the new. In addition, extensive use was made of applicable routines in NSWCLIB (the highly reliable NSWCDD Library of Mathematics Subroutines) in the development of STATLIB codes.

The first and only edition of STATLIB was published in August 1989 as an NSWCDD technical report.[10] STATLIB was designed for use on NSWCDD's general-purpose computer, and its code is, therefore, not transportable to other computers. STATLIB's programs perform I/O functions, and their codes are, thus, machine dependent. Initially developed for the CDC CYBER 170-865/875, STATLIB has been converted for use on subsequent NSWCDD general-purpose computers. These include the CDC CYBER 180-995E and the current Cray EL98 and Cray Y-MP2E computers. The decision to develop a library that was not transportable was based on the assumption that, apart from the members of the Mathematical Statistics Staff, most of its users would not be well-versed in statistical theory. Hence, it would be imperative that a program's results be organized and displayed in a clearly interpretable manner to be of maximum benefit to the analyst.

With the increased use of and dependence on personal computers (PCs) by NSWCDD scientists and engineers, it became apparent that the development of a PC version of STATLIB would be highly desirable. In 1989, Johnson began work on the development of a PC version. This work was completed in 1991 and was published in November 1991 as an NSWCDD technical report.[11] The resultant product, known as STATLIB/PC, contains the same programs and subroutines as the STATLIB mainframe version. The STATLIB/PC technical report refers the reader to the 1989 report for program descriptions and input guides. STATLIB/PC is executable on any IBM or IBM-compatible PC equipped with MS-DOS, Version 3.30 or higher. The programs in STATLIB/PC

were written in Ryan-McFarland Fortran.[12] The language used for the STATLIB/PC subroutines is ANSI Standard Fortran 77.

Contents of STATLIB

The 1989 version of the library contains approximately 27,000 lines of code. STATLIB comprises 34 programs for statistical data analysis and probability evaluation, and 24 subroutines for random-number generation. The programs are organized into seven distinct sections, and the subroutines are arranged in two sections. The headings of all sections are displayed in Table 7. The number of routines in each of the seven program areas varies considerably. This is primarily a function of the availability, at the time, of good alternative routines in existing commercial statistics packages. Generally speaking, if an existing commercial general-purpose computer statistics package that was locally available to NSWCDD scientists and engineers contained a reliable routine for performing a specific statistical analysis, the Mathematical Statistics Staff saw no reason to include a similar routine in STATLIB. Hence, STATLIB is not a complete library in the sense of a commercial package.

For example, STATLIB contains no programs for computing basic descriptive statistics or for categorizing or sorting data, because these facilities are available in existing packages. On the other hand, very little was available in existing packages for the computation of the power of statistical tests of hypotheses. The analyst who wished to determine the proper sample size for an experiment was usually forced to resort to published charts of power curves. Most of these charts contain families of power curves for different sample size values on the same graph. This, in turn, leads to difficulty in many cases in reading the correct value for sample size from the chart. The twelve power programs in STATLIB produce numerical tabular output, thus eliminating any problem associated with reading values.

In addition, STATLIB provides the analyst with the capability to compute power for most of the common statistical tests, as well as for some that are not so common. The sections on regression and goodness-of-fit analysis also contain a fair number of routines. The section on regression contains several programs with special features that help the analyst solve many of the problems encountered in regression analysis. Examples include routines for both uncorrelated and correlated weighted polynomial regression, and a routine for near-neighbor estimation of experimental error. The section on goodness-of-fit contains seven programs including one that allows the analyst to determine which distribution, if any, from the Pearson family of distributions best fits a set of data. This family contains distributions that are bell-shaped, J-shaped, L-shaped, and U-shaped.

Table 7. *Section Headings in STATLIB*

PROGRAMS
- **Regression Analysis**
- **Goodness-of-Fit Analysis**
- **Discrete Power Evaluation**
- **Continuous Power Evaluation**
- **Probability Evaluation**
- **Confidence Limit Evaluation**
- **Miscellaneous Statistical Analysis**

SUBROUTINES
- **Discrete Random-Number Generators**
- **Continuous Random-Number Generators**

The remaining three sections of programs in STATLIB contain specialized routines that the Mathematical Statistics Staff had employed on numerous occasions to solve problems arising in the NSWCDD scientific community. For example, routines to compute confidence limits for the Circular Probable Error (CEP) and Spherical Probable Error (SEP), to estimate the lethal dose 50th percentile (LD50), and to calculate binomial probabilities when the trial probabilities are variable were of recurring need to weapons analysts.

STATLIB contains 9 random-number generators for discrete probability distributions and 15 for continuous probability distributions. Since the use of Monte Carlo simulation was widespread within the NSWCDD scientific community, it was felt that the inclusion of a full range of random-number generators in STATLIB would serve the local community well. In addition to "standard" discrete and continuous generators, STATLIB included several that were not commonly available in other packages. Among these are discrete generators for an arbitrary (user specified) distribution and the uniform distribution (both with and without

replacement), and continuous generators for the uniform (both on a line and within a circle), lognormal, logistic, multivariate normal, Pearson, and three-parameter Weibull distributions. A routine that generates variates from a first-order Markov process was also included.

Usage of STATLIB

The STATLIB was heavily used by the Mathematical Statistics Staff in fulfilling its consulting and analysis mission, and it significantly improved the efficiency with which the group performed tasks requested by NSWCDD scientists and engineers. Outside of the Mathematical Statistics Staff, usage of the STATLIB within the NSWCDD scientific community was initially restricted to groups whose work required the use of NSWCDD's general-purpose computer. This computer was operated and maintained within the Strategic and Space Systems Department. In view of the fact that a few other technical departments at NSWCDD had procured their own mainframe computers (i.e., VAXs and DECs) to support major programs, STATLIB's lack of transportability limited its impact. Some attempts to convert portions of the STATLIB code for use on other mainframes within these departments were reported. Those groups that did utilize STATLIB did so on both classified and unclassified tasks. However, with the release of STATLIB/PC, usage of the library became more widespread. While no actual data is available regarding the frequency with which the library's routines have been used, comments and questions concerning the use of specific routines for project applications have provided a sense of its overall impact. Some of STATLIB's reported uses are given in Table 8. While the content of Table 8 is by no means complete, it does suggest the extent to which the library has been utilized and its impact on the NSWCDD scientific community. Those analysts and researchers who have employed the library as a resource tool have been highly complimentary of its utility and ease of use. The programs in STATLIB are initiated interactively by a menu driver that queries the user for the name of the program he desires to run. The user need only prepare an input file prior to executing a program. Since the subroutines are initiated within the user's main program, a listing of them does not appear in the menu driver.

Some requests for STATLIB's source code were received from outside NSWCDD after its

Table 8. *Reported Usage of STATLIB Routines at NSWCDD*

Type of Routine	Area of Application
Regression analysis	Prediction equations for propellant lot pressure Fitting equations to rocket motor thrust data
Goodness-of-fit analysis	Density estimation for missile fuze data and tactical computer performance characteristics
Power evaluation	Sample size determination for weapons accuracy studies
Binomial probability with unequal trial probabilities	Kill probability analyses for moving targets
Binomial confidence limits	Analysis of nonacoustic antisubmarine warfare (ASW) detection technique Reliability estimation for encased rocket data
Confidence limits for the CEP	Weapons accuracy studies
LD50 estimation	Explosive sensitivity analyses
2^k fractional factorial analysis	Missile performance studies
Random-number generation	Monte Carlo simulation

initial release in 1989. However, none of the requesting organizations had CDC mainframes, and no information is available regarding the success of any attempts to convert the codes. The release of STATLIB/PC in 1991 greatly facilitated the library's distribution. Several copies have been furnished to scientists and engineers within NSWCDD. However, demand for the PC version outside of NSWCDD has been sparse. This may be because the documentation for the PC version did not contain program descriptions and input guides, but rather, merely referred the reader to the earlier mainframe documentation for this information.

Relationship of STATLIB to Other Libraries

At the time of STATLIB's development, NSWCDD maintained two commercial packages on its general-purpose computer that contained statistics routines—the Statistical Package for the Social Sciences (SPSS-X) and IMSL. Since STATLIB's target audience was the NSWCDD scientific community, care was taken to minimize inclusion of routines that were already available in SPSS-X and IMSL. SPSS-X is a comprehensive tool for managing, analyzing, and displaying data. It is an excellent data management tool with many routines for performing basic statistical analyses. However, its treatment of specialized procedures such as nonlinear regression and nonorthogonal analysis of variance is limited. Although the majority of its routines are in the area of mathematics, IMSL does cover some of the specialized statistical areas that SPSS-X omits.

There were two additional highly regarded mainframe statistical packages available at the time—Statistical Analysis Software (SAS) and BMDP (Bio-Medical Data Package). SAS supported only IBM and other non-CDC mainframes and, hence, was not compatible with NSWCDD's general-purpose computer. Although BMDP leased a CDC version of its package, it was not purchased, because it had been found to be extremely time-consuming to install on NSWCDD hardware. Versions of SPSS-X, SAS, and BMDP for personal computers had also become available at that time. However, these early versions tended to be less comprehensive than their mainframe counterparts.

STATLIB was developed with the intention of providing statistical routines that were needed by NSWCDD scientists and engineers and were not locally available in existing commercial packages. The sections on discrete and continuous power evaluations are examples of badly needed facilities that were not available in other packages. Other routines, such as confidence limit evaluation for the CEP and SEP, which proved to be extremely useful to weapons analysts, were unique to the library. One area of significant overlap, however, is that of random-number generation. IMSL contains a fairly extensive set of such generators, but not all of those that are included in STATLIB.

Another objective of the STATLIB project was to develop programs whose output was well-annotated and contained all the necessary quantities that the analyst would need. In utilizing commercial statistical packages, members of the Mathematical Statistics Staff had observed that the output of certain programs could be difficult to interpret, especially by users who were not well-versed in statistical theory.

In addition, there were several instances in which routines failed to offer features that could be of added value to analysts and, especially, to statisticians in carrying out their consulting and analysis duties. It is believed that STATLIB achieved this latter objective and, hence, its programs generally produced more complete and more readily interpretable results than their existing commercial counterparts at that time.

In summary, STATLIB should be regarded as a specialized set of programs and subroutines that formed an excellent complement to the commercial statistical packages available at NSWCDD at the time of its release. The programs in STATLIB either did what the commercial packages did not, did it better, or simply allowed the user to have more control over the analysis.

Availability of STATLIB

STATLIB will continue to be made available to all interested users on NSWCDD's general-purpose computer, and its source code will be freely shared upon request. Copies of STATLIB/PC are also available upon request. All requests concerning STATLIB and STATLIB/PC should be directed to Dr. John R. Crigler at

540-653-7601. I also serve as the technical point of contact for questions concerning usage of STATLIB and STATLIB/PC.

Concluding Remarks on STATLIB

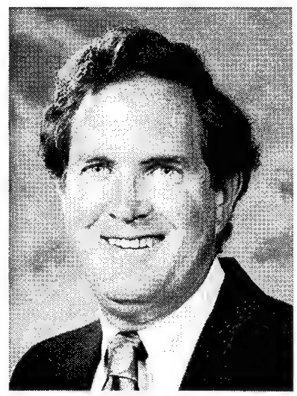
The STATLIB building project achieved its dual objectives of improving the consulting and analysis capabilities of the Mathematical Statistics Staff, and providing a valuable resource tool to the entire NSWCDD community. Unique capabilities included in STATLIB helped to fill some of the voids in locally available commercial packages at the time. After STATLIB's release, there were a few requests to develop and incorporate additional routines in a future edition. At that time, however, the Mathematical Statistics Staff could no longer afford to dedicate the manpower required to further the development of the library.

Since STATLIB's release, retirements and job transfers have resulted in the dissolution of the Mathematical Statistics Staff. At the present time, I am the only remaining member of that group who continues to provide statistical consulting and analysis services at NSWCDD. Consequently, there are no current plans to enhance the library's capabilities. However, NSWCDD is committed to ensuring that the library is maintained on its general-purpose computer. As a result, it is anticipated that STATLIB will continue to be a valued resource within the NSWCDD scientific community in the foreseeable future.

References

1. Morris, A. H., Jr., *Development of the NSWCDD Mathematics Subroutine Library,* unpublished notes, Dec 1992.
2. Morris, A. H., Jr., *NSWCDD Library of Mathematics Subroutines,* NSWCDD TR 3843 (DTIC), Jun 1978, NSWCDD, Dahlgren, VA.
3. Morris, A. H., Jr., *NSWCDD Library of Mathematics Subroutines,* NSWCDD TR 79-338, Mar 1980, NSWCDD, Dahlgren, VA.
4. Morris, A. H., Jr., *NSWCDD Library of Mathematics Subroutines,* NSWCDD TR 81-410, Oct 1981, NSWCDD, Dahlgren, VA.
5. Morris, A. H., Jr., *NSWCDD Library of Mathematics Subroutines,* NSWCDD TR 84-143, Jun 1984, NSWCDD, Dahlgren, VA.
6. Morris, A. H., Jr., *NSWCDD Library of Mathematics Subroutines,* NSWCDD TR 86-251, Apr 1987, NSWCDD, Dahlgren, VA.
7. Morris, A. H., Jr., *NSWCDD Library of Mathematics Subroutines,* NSWCDD TR 90-21, Jan 1990, NSWCDD, Dahlgren, VA.
8. Morris, A. H., Jr., *NSWCDD Library of Mathematics Subroutines,* NSWCDD TR 92-425, Jan 1993, NSWCDD, Dahlgren, VA.
9. Ward, Robert C., "Numerical Computation of the Matrix Exponential with Accuracy Estimate," *SIAM J. Numerical Analysis 14* (1977), pp. 600-610.
10. Thomas, M. A.; Gemmill, G. W.; and Crigler, John R., *STATLIB: NSWCDD Library of Statistical Programs and Subroutines,* NSWCDD TR 89-97, Aug 1989, NSWCDD, Dahlgren, VA.
11. Johnson, Gary M., *STATLIB/PC Users Guide,* NSWCDD TR 91-549, Nov 1991, NSWCDD, Dahlgren, VA.
12. Ryan-McFarland Fortran (RM/FORTRAN™), Version 2.4, AUSTEC Inc., Rolling Hills Estates, CA, 1987.

The Author



JOHN R. CRIGLER is a Mathematical Statistician in the Advanced Computation Technology Group of the Systems Research and Technology Department. He received a B.S. degree in mathematics from Washington and Lee University in 1968, an M.S. degree in statistics from Virginia Polytechnic Institute and State University (VPI&SU) in 1970, and a Ph.D. in statistics from VPI&SU in 1973. He joined NSWCDD in 1972 and has previously worked in the Warfare Analysis Department (now the Strategic and Space Systems Department) and the Engineering and Information Systems Department providing statistical consulting, analysis, and research services to the entire NSWCDD community. He is currently working in support of the system evaluation and assessment technology task of the Engineering of Complex Systems project by developing an instrumentation-based simulation model for predicting the resource requirements of future complex systems. He is a member of the American Statistical Association, the Biometric Society, and the Institute of Mathematical Statistics. In addition, he has several technical publications to his credit.

Engineering of Complex Systems

Harry E. Crisp, II

The U.S. has a proud heritage of developing classic examples of large-scale, complex, mission-critical systems. These systems have been obtained at a considerable investment of resources, including the managers, scientists, and engineers who spearheaded the development; and the operators and maintainers who provide successful use and life support. Future versions of existing systems, as well as conceptual new systems, are envisioned to be based on much advanced technology baselines, thus enabling dramatically different system architectures. Much-improved performance capabilities are anticipated from these new architectures in addition to shorter development schedules and reduced costs. More efficient and effective use of human resources is also anticipated, both in the development of the system and in its operational use. In order to achieve these objectives, an integrated design methodology is needed that enables a seamless flow across the system development process as well as "flow-down" into the application specialties.

Introduction

Many systems in industrial and commercial applications, including military, might be described as "complex."[1] The modern automobile is a good example of a complex system with extensive use of computers and sensors to control engine operation, automatic transmissions, passenger compartment air temperature, antilock braking systems, safety systems, security systems, stereo systems, etc. Other examples of complex systems commonly used by everyone in our modern society include the electric utility systems, telephone systems, commercial aircraft, banking systems, reservation systems, personal computers, and the Internet. The rapid evolution of electronics and computing technology has led to extensive automation of nearly all consumer products and the manufacturing processes that produce them. The application of digital computer technology is pervasive, to the point that we take for granted that computer chips are embedded in many household appliances.

All too often, we also take for granted that we can engineer and produce highly complex systems. Certainly, we as a nation have a significant record of achievement in fielding such systems as the Apollo moon shots, the Space Shuttle, our vast network of telecommunications systems, the strategic missile triad, high-performance military aircraft, sophisticated commercial aircraft, TRIDENT nuclear submarines, and the AEGIS cruisers and destroyers. Overlooked quite often, is the tremendous investment in resources required to accomplish these feats, not the least of which was the investment in scientific and engineering talent.

As we look to the future and envision the next generation of complex systems, we recognize that the continued evolution of many technologies (electronics, computing, telecommunications, materials, sensors, propulsion systems, etc.) will enable the development of systems that are greatly advanced in terms of performance. One anticipates also the likelihood that these future systems will be much less manpower intensive to operate, more energy efficient in operation, less harmful to the environment, and

less expensive to build and maintain. The challenge, likewise, is to learn from the lessons of the past in the engineering of these systems and to utilize technology advances effectively to improve the practice of engineering large, complex systems so as to achieve these goals.

Trends

A number of ongoing trends have the potential to significantly impact the practice of the engineering of systems.[2-4] These may be classified as technological, geopolitical, economic, or acquisition policy. Technological impacts principally stem from the rapid advancement of various technologies, particularly computing and telecommunications. Technology evolution provides the opportunity to advance the performance of the intended product. The challenge is to make wise decisions about which technology solution is the most appropriate for the problem at hand. Such a decision-making process requires a methodology (process, methods, and tools) for defining the requirement, identifying the potential solutions, and performing tradeoff analyses. Technology evolution also has the potential to enable a methodology to improve the practice of determining the appropriate technology baseline for new systems. The evolution of computer-aided engineering environments, coupled with simulation-based design techniques, virtual prototyping, and collaborative engineering capabilities, are examples of the components of an improved methodology.

Geopolitical impacts have a particular significance to our military systems. The end of the Cold War has reduced the apparent need for major new military systems. Additionally, many of the older systems in the inventory have been retired. On the other hand, the systems that remain in the inventory are confronted with the need to address new missions and new types of threats. Limited-conflict engagements place a priority on rapid deployment, flexible use of resources, and surgical strike capabilities. Operation in littoral regions, as opposed to open ocean, provides a particular risk to Navy combatants. Hence, existing systems need to be reengineered for capabilities to address the new missions and threats. Envisioned new systems will need to be designed to meet these missions and threats with improved levels of performance, yet be less expensive and more quickly built and deployed.

Economic impacts are occurring across a broad front—the first of these is the downward trend in the defense budget resulting from the end of the Cold War. Naturally, this resulted in the retirement of older systems as well as the slowdown of new systems acquisition. Another consequence is a "shaking out" of the companies involved in the defense business. This has taken the form of a number of mergers and restructuring of corporate organizations. A number of companies have moved away from traditional military product lines towards consumer products. Global competition has also impacted corporate strategies with regard to product lines and supporting resources. Downsizing and rightsizing has resulted in large layoffs of technical staff and early retirements of many senior technical experts. Many of the traditional engineering teams no longer exist. The impact is also felt in the government research and development laboratories where ongoing reductions in staff and hiring freezes have resulted in significant losses of senior personnel and very little hiring of new college graduates. Although new methods and tools made possible by technology advances can significantly increase productivity of the remaining scientific and engineering staffs, the loss of the knowledge base gained by hard experience is difficult to overcome.

Acquisition policies are also undergoing many changes stimulated by the need to acquire systems more efficiently and at less cost. Consequently, many military standards are being eliminated to be replaced by performance specifications and commercial standards. The use of commercial off-the-shelf equipment is also preferred, wherever possible, as opposed to standardized military equipment—especially in electronics, computers, and communications. Electronic systems are now to be acquired based on "open-system" specifications in order to facilitate the use of commercial equipment and later upgrades. All of these changes can significantly reduce the initial system acquisition costs, but long-term, life-cycle costs may be adversely affected by the typical rapid turnover in commercial product lines. Accessibility to design information is also a typical problem in using commercial products in mission critical systems

where complex interactions and safety considerations demand predictable performance.

The consequence of the foregoing is that a significant challenge currently exists to improve the practice of engineering of complex systems so as to accomplish the goals of "better, cheaper, quicker" in next-generation system acquisitions.

Complex System Characteristics

The systems of primary interest are those that might be described as large-scale, complex, real-time systems. Some characteristics[5] of these systems include:

- The use of technologies and solutions that cross multiple application domains (e.g., sensors, weapons, communications)
- A high degree of system automation through the use of large-scale embedded computer systems (generally several million lines of executable code) that:
  - Use many heterogeneous resources that may be configured in distributed or highly parallel multiprocessor architectures
  - Respond to hard, real-time processing requirements driven by rapidly changing environments and conditions
  - Operate in multiple operating states and modes
- The need for continuous, reliable, fault-tolerant operation
- The demand for safety and security requirements

These systems typically require the collaborative efforts of multicontractor teams and, sometimes, multiple teams. They can take over ten years to develop and have a service life of thirty-plus years. During the life cycle of these systems, they will be periodically upgraded to maintain their capability and keep pace with technological advances. This places an emphasis on an "evolutionary acquisition process" that can lessen the impact of baseline changes and modernizations. System architectures that provide flexibility and ease of change are also essential.

Many systems throughout industry and the military possess one or more of the foregoing characteristics. However, *mission critical systems* tend to possess all of these characteristics simultaneously. This creates conflicting requirements that challenge the ingenuity of the design team to resolve. There also is a tendency toward nonlinearity, nondeterminism, and mathematical intractability due to the large number of interfaces, massive amounts of data flow, and complex interactions that occur among the processes embodied within these systems. The result is that a mix of methods must be utilized to engineer the system, including expert opinion and heuristic techniques in addition to best practices and traditional engineering methods.

The AEGIS combat system[6] (illustrated by Figure 1) provides a useful example of the characteristics discussed above. Each of the indicated combat system elements are in and of themselves "complex systems" since they typically require multiple technologies to implement; are highly automated and include embedded computing resources; include real-time constraints; must be reliable, safe, and secure; and have multiple operating states and modes. It is also true that each of them typically represents an application domain (e.g., sensor, weapon, communications, navigation, control, etc.) for which there is a substantial body of technical knowledge and supporting culture. However, no one of these individual elements is sufficient to accomplish the various warfare missions of the ship. It is necessary that they each function in consonance with other elements in the combat system to accomplish a single mission such as antiair warfare (AAW). The primary elements required to accomplish AAW in the AEGIS combat system are indicated in Figure 1. Viewed in this fashion, it is obvious that a "system of systems" configuration with significant interfacing, data flow, and control flow is required to accomplish AAW. When we add antisubmarine warfare (ASW) as a simultaneous mission requirement, we see that some degree of sharing of combat system assets is required to accomplish the two missions simultaneously. This requires command-level coordination, conflict resolution, and resource allocation strategies that are likewise supported by extensive interfaces, data flow, and control flow between command, warfare control, and combat-system elements. When other simultaneous mission areas are added, such as antisurface warfare (ASUW), strike warfare (STW), and electronic warfare (EW), the full significance of the engineering challenge for this hierarchical

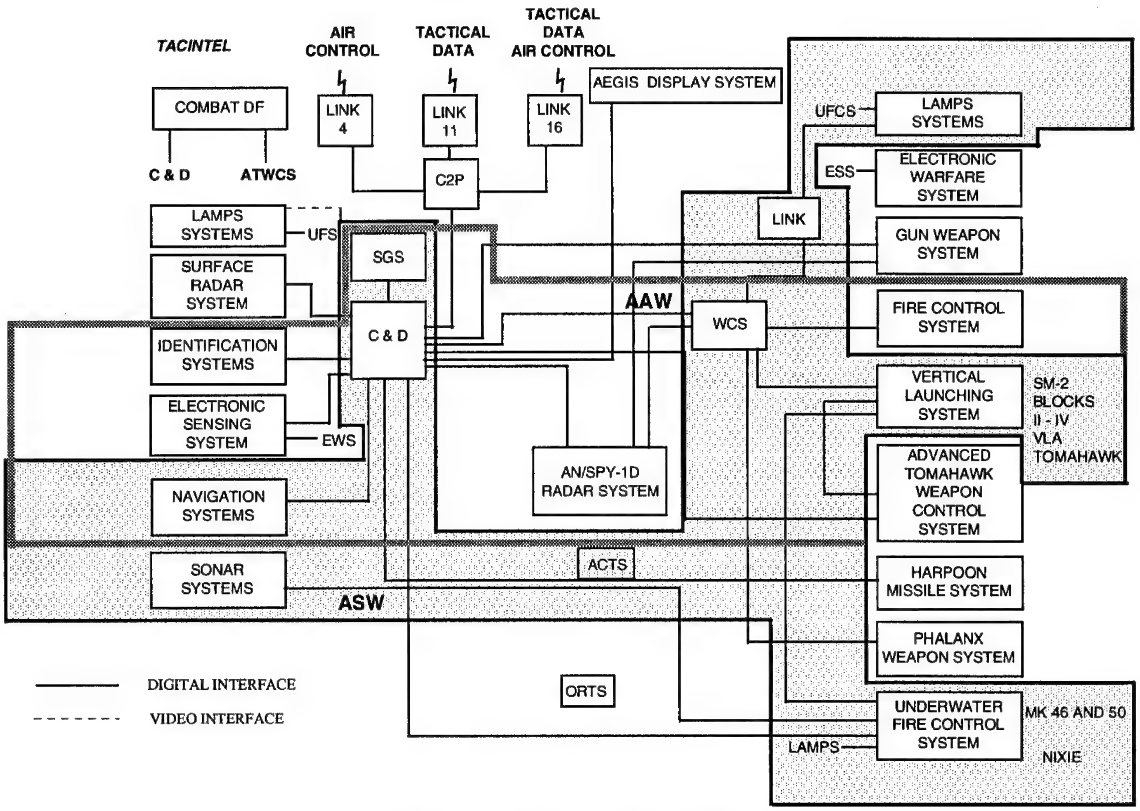


Figure 1. *AEGIS Combat System.*

"system of systems" is realized. Further, this is a single ship example; task force operations and joint warfare operations[7] increase the scale of complexity exponentially.

Examples of other systems that feature characteristics of large-scale complexity such as represented by the AEGIS system exist in a variety of industrial and commercial applications. However, the distinguishing feature of military combat systems is the fact that a diverse array of weaponry is involved. Along with this is the significant role played by human operators and decision- makers. Many of the tasks performed by operators and decision makers are unique cognitive and physiological capabilities that are very difficult to automate, yet are essential in achieving system functionality and performance. Other tasks are not desirable to automate due to all the many implications of whether or when to fire weapons.

The Problem

Future versions of large, complex systems will be based on a much more advanced technology baseline than that implemented in currently deployed systems. In particular, continually evolving electronics and computing technologies provide the basis for dramatically new approaches to architecting these systems. Yet, little is available in the form of an integrated methodology[8] for addressing the engineering of complex, long life cycle systems. Existing engineering processes, methods, and tools are typically unique to a given problem domain and do not support the full system development and deployment cycle. Nothing is readily available to address the complex tradeoffs between hardware, software, and human elements of the system. Little is available to support the formal assessment of the design against the original requirements.

Engineering complex systems requires the capability to manage and analyze large amounts of data; to account for complex interactions between the hardware, software, and human elements within the system; and to apply appropriate domain knowledge inherent to the problem to be solved. To do so requires the application of argumentative (expert opinion) and heuristic (rule of thumb) methods as well as normative (best practices) and rational (science and math) methods.[9]

There are a number of technical issues that must be addressed to accomplish the needed integrated methodology. Among these are:

- Systems Architecting: There is a lack of formal means supported by automated capabilities to fully explore system architecture options and to incorporate all the stakeholders views concerning the system requirements.
- Design Capture: Current capabilities to capture, analyze, and evaluate full system designs are inadequate to support the full range of design tradeoffs and do not support all phases of the engineering process.
- System Assessment: System-level metrics, instrumentation, and evaluation techniques to measure and predict system design attributes at every stage of development are weak.
- Computer-Based Systems: Inadequate methods exist for integrated codesign of hardware, software, and human tasks for complex systems. A common computer systems lexicon across all application domains does not exist.
- Infrastructure and Tools: Currently available automated capabilities are limited to hardware or software applications, do not extend to system level requirements, are not scalable to large-scale problems, and are not interoperable.
- Systems Engineering: There is no broadly accepted systems engineering process, hence, little basis for a common lexicon or representation languages. This, in turn, impacts the feasibility for tool vendors to develop automated capabilities that can fully support all phases of the engineering process.
- Domain Engineering: Better processes and methods are needed to support design reuse within application domains. A common lexicon for computer-based systems that supports interactions across domains is also needed.
- Reengineering: Capabilities to recapture original design rationale and transform to new architectures are weak. System architectures that support a proactive approach to reengineering are needed.

The Engineering of Complex Systems (ECS) Technology Project

The foregoing list of technical issues motivated the Naval Surface Warfare Center, Dahlgren Division (NSWCDD) in 1990 to propose the initiation of an exploratory development program to the former Office of Naval Technology (now consolidated with the Office of Naval Research). The ECS Technology Project addresses the underlying technologies required to achieve an integrated methodology for the design and integration of large-scale, complex systems. The goal is to enable design teams to synthesize and evaluate system design options for mission-critical requirements with orders-of-magnitude improvements in productivity and accuracy over conventional methods. This, in turn, will provide high potential for achieving needed warfighting requirements and mission capabilities while simultaneously reducing cost and scheduling risks.

The approach taken by the ECS Technology Project focuses on technical tasks in systems design synthesis, evaluation and assessment, and reengineering and reuse. These are addressed in the context of a generic, forward engineering process and a reengineering process. Figure 2 depicts the phases of a typical forward engineering process, with feedback arrows indicating the usual iterative nature of the process. Also indicated are the areas of technology effort being conducted by the ECS Technology Project.

Systems Design Synthesis Task

The ECS Systems Design Synthesis Task encompasses efforts in requirements specification, design capture, design structuring, and resource allocation. The intent is to provide an integrated environment that facilitates a seamless flow across the phases of the forward engineering process. "Flow-down" from the overall systems level to specific application domains (e.g., combat system elements) of key system design information (top-level requirements, specifications, architectures, etc.) must also be enabled.

The objective of the requirements specification efforts, for example, is to support a multidisciplinary design team with a design environment that provides an interface that

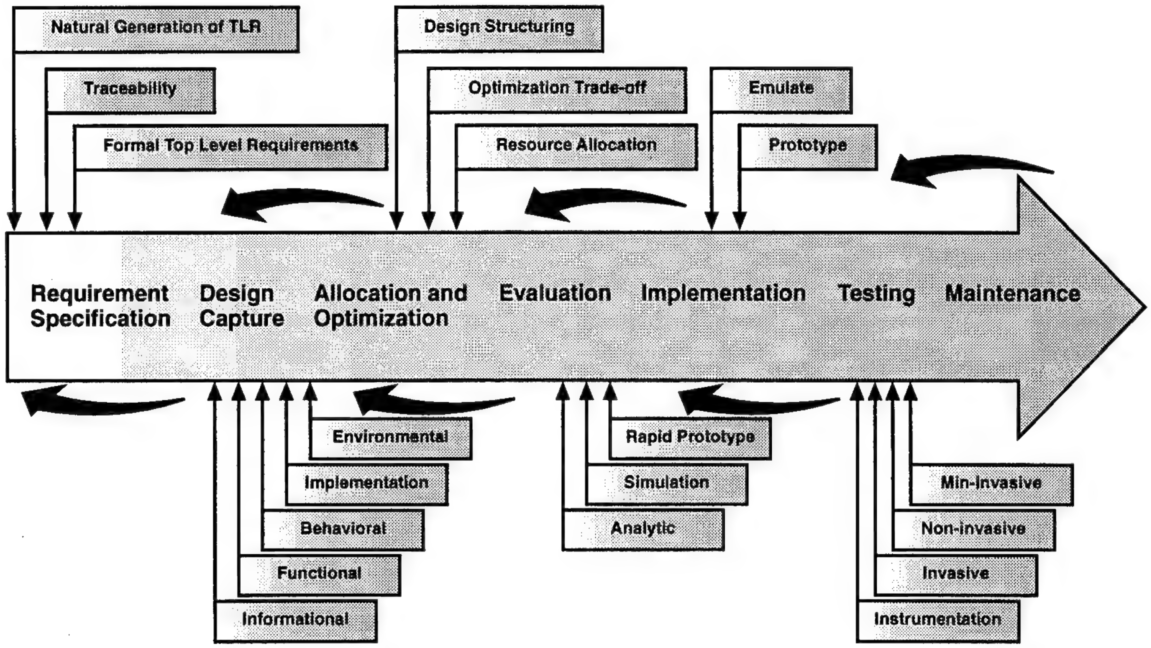


Figure 2. *ECS Forward Engineering Process.*

enables each of the design team members to interact with the design record in a manner "natural" to their area of expertise. If the problem is to design an automobile, the chassis, body, power train, electronics, and interior design experts must be able to interact with the total design as it evolves. A natural interface with an automated design repository can significantly increase productivity and assure consistency and completeness. This needs to be coupled with methods that assure traceability of the original requirements throughout the design process and the application of techniques for formal specification of requirements.[10] These provide a basis for evaluation of the product against the original specifications.

Figure 3 depicts the *multiple view design capture* process[11] being explored within the Systems Design Synthesis Task. The information view (system entities, relationships between

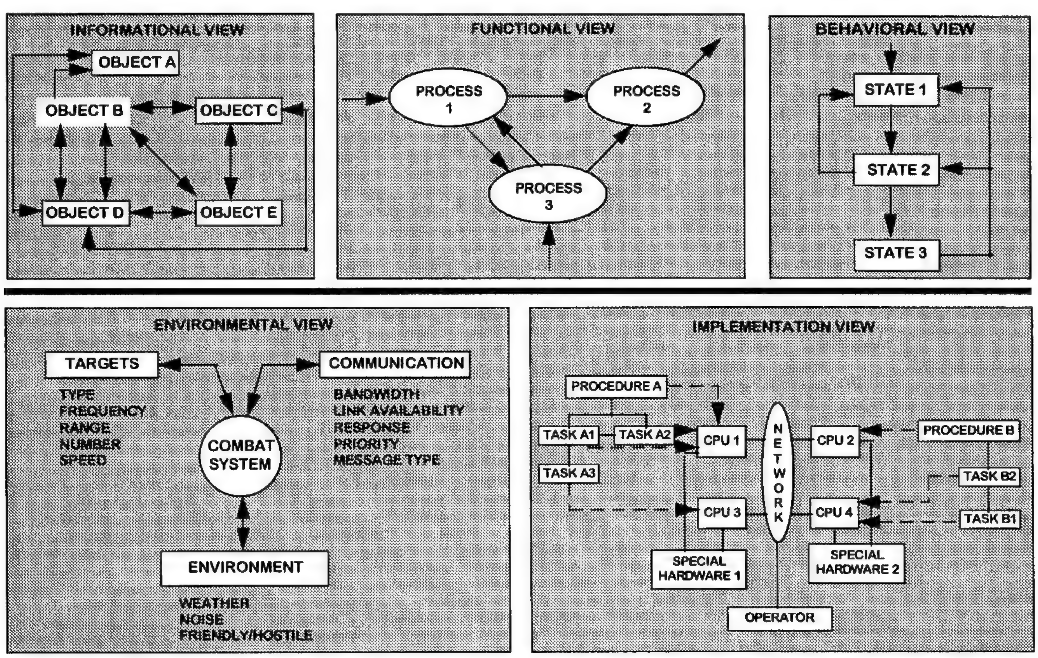


Figure 3. *ECS Multiview Design Capture.*

entities, and attributes), functional view, and behavioral view are reasonably well supported by currently available commercial tools. However, the implementation view (candidate hardware, software, and human operator resources) are not currently well-supported. This is also true of the environmental view. Prototype capabilities to support these views are under development within ECS.

An additional objective of the ECS multiview design capture effort is to establish a high degree of interoperability among automated capabilities that support these views. This will enable the implementation of an integrated environment for a seamless flow between the various design views, allowing the designer to access the design view most appropriate for each phase of the design process. The mechanism for accomplishing this is the *System Engineering Technology Interface Specification* (SETIS)[12] under development by ECS to support the exchange of both data and semantic information.

A third area of activity within this task is *design structuring, resource allocation, and optimization*[13] as depicted in Figure 4. This effort explores capabilities to support partitioning the design, strategies for allocating resources, and optimization techniques. A key prototype—*Destination*—guides the designer through a set of design factors[14] such as performance, safety, security, reliability, maintainability, extensibility, usability, cost, etc. Based on acceptable bounds for these factors established by the designer, *Destination* provides interactive support to guide the designer through alternative system implementations. A library of optimization algorithms can be accessed to evaluate how well each alternative satisfies the range of design factors.

Systems Evaluation and Assessment Task

The increasing complexity of naval systems makes it imperative that early and continual evaluation and assessment be performed to assure optimal design. Figure 5 provides a conceptual view of an integrated environment needed to achieve continual evaluation from the earliest stages of the design process through operational evaluation. It is based on implementing an infrastructure to integrate and support a system design repository, system design analysis capabilities, and a system evaluation and assessment repository. Such an

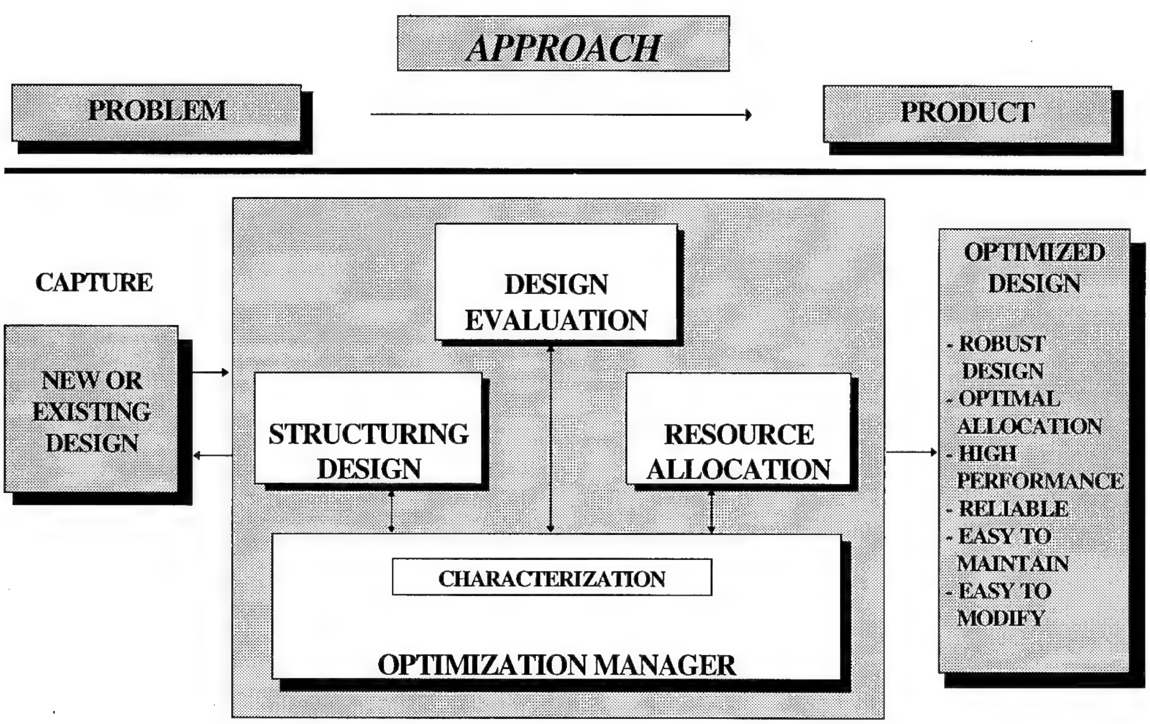


Figure 4. *System design structuring, resource allocation, and optimization.*

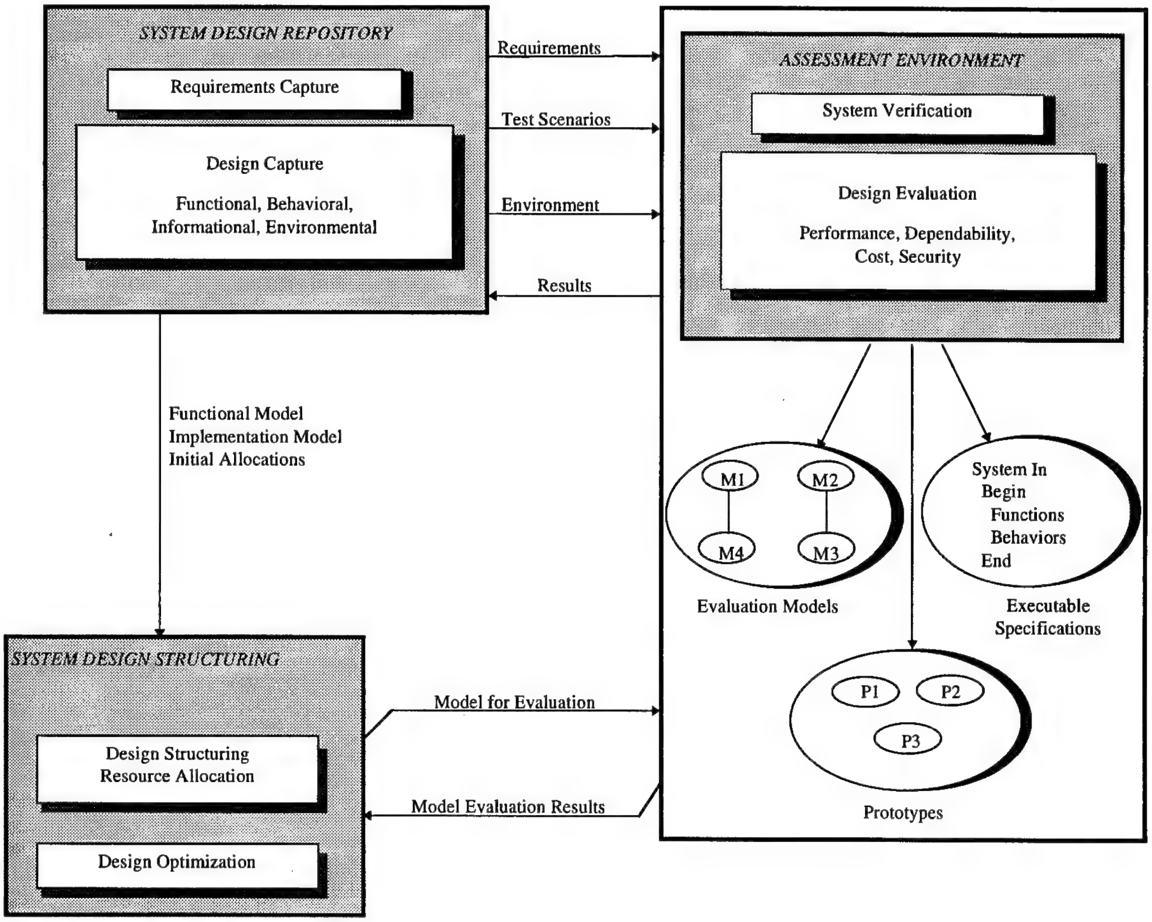


Figure 5. *System assessment environment.*

environment also provides a baseline for the system life support environment.

The evaluation and assessment capabilities required to accomplish the assessment environment depicted in Figure 5 require a very active program in *modeling, measurements and instrumentation, virtual prototyping, and infrastructure and repositories*. It will employ emerging technologies such as multimedia, intelligent agents, virtual reality, and object-oriented technology to ensure the generation and application of timely modern products.

Modeling technology produces the quantitative and qualitative results needed to determine viable alternatives and perform tradeoffs among such alternatives. A variety of models are required (analytical models, discrete-event simulation models, dynamic performance models, prototypes) at various stages of design to evaluate critical system attributes such as cost, schedule, funtionality, performance, reliability, availability, maintainability, safety, security, and usability. ECS project technical efforts are developing a "meta-model" for leveraging available modeling technology to support the design evaluation process. Interoperability between large-grain (e.g., force level) and fine-grain (e.g., sensor-actuator) models is also an area of interest.

Technology efforts in measurements and instrumentation provide metrics for all aspects of system development including metrics on products, processes, components, and system life cycle. In addition, technology is addressed that enables designers to observe and evaluate system behavior in the least intrusive manner possible. This includes three overlapping categories of instrumentation: invasive, minimally invasive, and noninvasive. Key areas of work are:

1. Identification and validation of key metrics for modeling and assessment
2. Extension of the existing software reliability program[15] to address system reliability

3. A performance modeling capability[16] driven solely by metrics collected via computer instrumentation
4. An integrated measurements framework[17] for total system evaluation and assessment using metrics

Key prototypes in process include the Statistical Modeling and Estimation of Reliability Functions for Software (SMERFS), Metrics Information System Tool (MIST), and the Visual Simulation Environment (VSE).

Advancements in computer simulation and visualization technology provide an opportunity for systems engineering technology to benefit from *virtual prototyping*, or the application of virtual reality technology to system engineering.[18] This involves the creation and exercise of prototypes of a system that, when driven by scenarios representing the anticipated real environment, will allow participants to experience sensory immersion. Virtual prototyping makes it possible to address issues that are difficult to evaluate effectively other than through user and developer sensory immersion into a synthetic environment representing the system under development. This approach is effective in obtaining customer input in the early stages of design on the degree to which alternate designs satisfy intended missions and design goals. It is also effective in evaluating the desired role of operators and decision-makers in the system and in obtaining user feedback on operability aspects of the design. Virtual prototyping is an area of planned future technical effort within ECS.

Systems Reengineering and Reuse Task

Figure 6 depicts the generic ECS systems reengineering and reuse process intended to support long life-cycle (evolutionary) systems. The thrust of the ECS effort is to enable an active approach to reengineering and reuse (versus reactive). That is, reengineering and reuse should be planned for in the initial design of the system. This requires the establishment of complete design records, repositories of reuse components, and system architectures that support design evolution. Key steps in the reengineering process include recapturing the design of legacy systems (including the original design rationale), placing the design in a level of abstraction that enables analysis and restructuring, and transforming the design to meet new requirements.

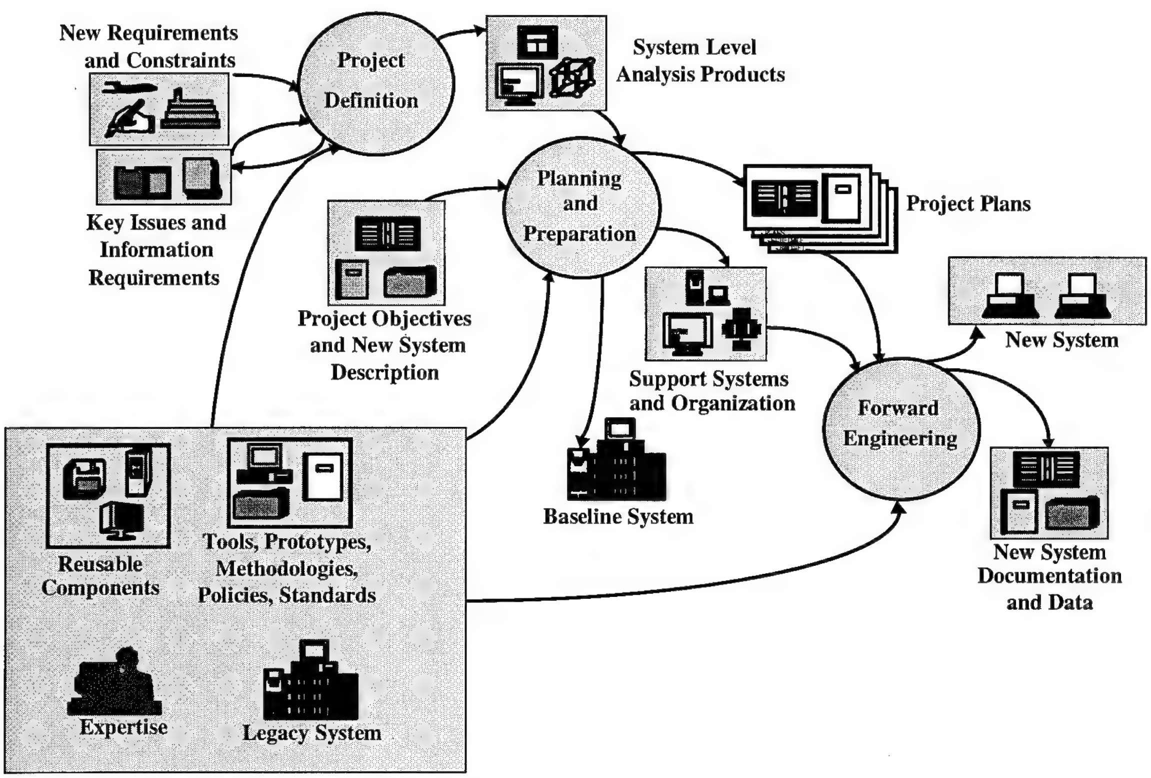


Figure 6. *System Reengineering and Reuse.*

A significant problem for many current Navy systems is the reengineering of existing tactical software programmed in CMS-2 and hosted on obsolete processing architectures. These programs need to be translated to Ada and transformed to modern processing paradigms. A key ECS prototype, the Software Reengineering Environment (SRE), has been developed to address this problem. The SRE accepts CMS-2 source code, parses it, and translates it to an intermediate representation language that enables graphical presentation of the program structure and provides interactive support to the software engineer in optimizing the Ada program.

The Advanced Research Projects Agency (ARPA) and the Joint Logistics Commanders (JLC) have cofunded the integration of the SRE with a software engineering environment (SEE) under development by Boeing to provide a full capability for reengineering and reuse.[19] Figure 7 depicts the integrated environment, with the Boeing SEE providing domain engineering and a repository of software reuse components. The SRE provides the facilities for reengineering of existing software to stock the repository, as well as software understanding and software specification assistance. This integrated environment will be used in a demonstration project during 1996 that will address flight simulator software at the Naval Air Warfare Center Training Division. It will also be used in a demonstration project at the Army Missile Command in Huntsville, Alabama, that will address missile guidance and control software.

Workshop on Engineering of Systems in the 21st Century (WES21)

The WES21 series of annual workshops[20] was established under the auspices of the ECS project in an effort to form a collaboration among industries, universities, and the Navy in addressing key needs in engineering next-generation systems. It is cosponsored by the Chief of Naval Research, the Naval Surface Warfare Center, and the Naval Command, Control, and Ocean Surveillance Center. A goal of WES21 is to assure a wise investment in systems technologies. The first WES21 was conducted June 28 through 30, 1994, at the Sheraton Inn

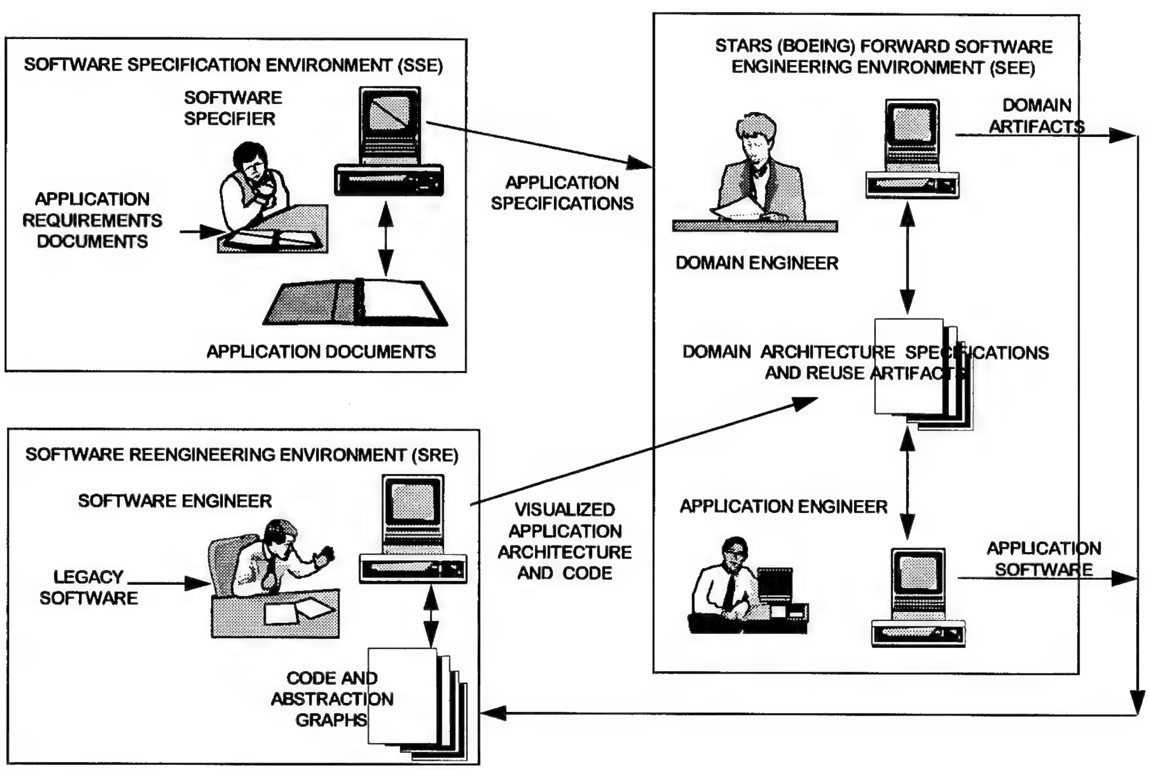


Figure 7. *Integrated SRE and SEE Environment.*

in Fredericksburg, Virginia. Over 120 prominent system engineering practitioners from industry, government, the services, and academia were invited participants. The second WES21 was conducted June 12 through 14, 1995, at the University of Maryland Conference Center with over 175 invited participants. The results of each WES21 have been documented in a proceedings.[21,22]

The participants for each of the WES21 workshops were divided into focus groups to address specific topics, such as:

- Computer-based systems implementation
- Design capture
- Domain engineering
- Evolutionary systems
- Infrastructure and tools
- Organizational/institutional learning
- Program management of complex systems acquisition
- Reengineering
- Revolutionary systems
- Standards/metrics/quality
- Systems architecting
- Systems assessment
- Systems engineering management

They were asked to answer questions such as *Where are we today? Where do we want to be? What are the obstacles to getting there? How do we get there?*

WES21 Findings

A common theme that emerged from all the WES21 focus groups is the need to anticipate in order to shape the future environment within which complex systems are engineered, as well as to communicate a clear view of what such an environment should be across the entire engineering community. The perspective of the focus groups was that the fundamental responsibilities of the engineers creating the complex systems envisioned in the 21st Century will not change. They will continue along the natural path of engineering—identifying and implementing the best technological solutions to meet the user's requirements. The single most important requirement for these talented engineers is the need to recognize the broad conceptual architectures—the systems of systems—that are emerging today within the defense and commercial sectors. Engineers can no longer simply understand their own system or function, such as mechanical, electrical, or software engineering, but they must also understand how their system or function fits in the overall engineering enterprise. Compounding this challenge is the fact that engineers will find themselves under increased pressure to achieve these broad-based solutions effectively—in the shortest period of time with a reasonable expenditure of funds.

The WES21 focus groups identified numerous opportunities to shape the future environment for engineering complex systems. These can be grouped into three common areas: meta-process, integration, and education.

Understanding the *meta-process for engineering complex systems* is viewed as the key to success in the future, along with establishing mechanisms for communicating and documenting the process and resulting product. A meta-process describes the steps to be taken and the information to be captured and generated at each step in the development of a complex system. The meta-process is described in executable models and is independent of particular views, notations, and methodologies. It may be tailored to the methodology in favor at a particular institution. It also provides the basis for guidance with regard to needed full-spectrum capabilities in automated engineering tools and environments.

Since everyone involved in the creation of a complex system needs to understand the process and their role in it, an important part of the process is creating strong communications throughout the team. Rapidly emerging computing and telecommunications technology provides an opportunity to mitigate the issues related to loss of corporate knowledge and overcome the challenges faced by noncollocated design teams.

The increased demand for quality, along with heightened performance requirements, calls for *tighter integration among tools* than ever provided before. A test bed for integrated tools is envisioned to advance the tools available to automate engineering of complex systems, demonstrate tool operability, and lead to automated exploration of systems solutions in a world where reuse is frequent or dominant.

There was a common recommendation to provide *education and training* to all stake-

holders including policymakers, users, program managers, engineers working the solution, and student engineers. A strong need exists to train individuals and teams in the engineering of complex systems best practices (the meta-process) and in the use of an integrated tool set. The recommended tools for learning ranged from using test beds to demonstrate successful approaches for engineering complex systems, to increasing the use of modeling, simulation, and multimedia training to expand understanding at all levels. Equally important is the need to provide a clear understanding of the process at all phases of the system development.

WES21 Objectives for the Future

WES21 activities during the past three years have provided results in terms of increased communication among the various technical communities involved with the engineering of complex systems, more tightly coupled research efforts, and increased awareness of the challenges to be solved. If the focus groups' vision for the 21st Century is to be achieved, the next several years require focused, consistent progress toward defined goals. WES21's major objectives for 1996 through 1998 are listed below:

1. Broad recognition of the scientific basis for science and technology investment in the engineering of complex systems
2. Development of a science and technology investment strategy based on the focus groups' recommendations that is recognized by DoD, industry, and academia as addressing broad needs in the engineering of complex systems
3. Establishment of WES21 as the "umbrella" forum for bringing together a coalition of various professional societies and industry associations involved in technical activities underlying the engineering of complex systems
4. Establishment of funding support from government and industry for a broad range of continuing activities in the engineering of complex systems, such as information databases, tool repositories, and research tasks

Conclusions

There is a significant need for new processes, methods, and tools to support the engineering of the next generation of large-scale, complex, mission-critical systems. This need is becoming recognized throughout industry, academia, and government. The ECS Technology Project, sponsored by the Office of Naval Research, is addressing key needs in systems design synthesis, systems evaluation and assessment, and reengineering and reuse. Prototypes of advanced automated capabilities have emerged from the ECS efforts and are available for evaluation. The Workshop on Engineering of Systems in the 21st Century has succeeded in recruiting experienced practitioners, researchers, and managers from across the community to collaborate in identifying the state-of-practice, ongoing trends, and needed investments. A long-term objective will be to gain broad recognition of the need for a strong science and technology investment to advance the state-of-practice in the engineering of complex systems.

References

1. White, Stephanie, et. al., "Systems Engineering of Computer-Based Systems," *Computer*, Vol. 26, No. 11, Nov 1993, pp. 54-65.
2. Lake, Jerome G., "Systems Engineering Re-Energized: Impacts of the Revised DOD Acquisition Process," *Engineering Management Journal*, Vol. 4, No. 3, Sep 1992, pp. 8-14.
3. Fisher, David A. and Skolnick, Alfred, "Acquisition Reform and Best Product Procurement: An Engineering View," *Naval Engineers Journal*, Vol. 106, No. 6, Nov 1994, pp. 41-57.
4. Geary, John W. and Masters, Michael W., "Investigating New Computing Technologies for Shipboard Combat Systems," *Naval Engineers Journal*, Vol. 107, No. 3, May 1995, pp. 253-265.
5. Harrison, Robert; Laplante, Phillip A.; and Stoyenko, Alexander D., "Doubling the Engineer's Utility," *IEEE Spectrum*, Vol. 31, No. 12, Dec 1994, pp. 32-39.
6. Stein, K. J., "AEGIS System Tested Successfully," *Aviation Week and Space Technology*, Vol. 102, No. 13, 7 Apr 1975, pp. 36-40.
7. Owens, William A., "The Emerging System of Systems," *Proceedings of the U.S. Naval Institute,* Vol. 121, No. 5, May 1995, pp. 35-39.
8. Alford, Mack and Malcolm, Nadine E., "Systems Engineering: Its Roots, Its Future," *CASE Trends,* Oct 1993, pp. 57-61.
9. Rechtin, Eberhardt, "The Art of Systems Architecting," *IEEE Spectrum,* Vol. 29, No. 10, Oct 1992, pp. 66-69.
10. Harrington, Gail; Ramash, Bala; and Rondeau, Kathleen, *A Model of Requirements Traceability to Support Systems Development,* Naval Postgraduate School Technical Report NPS-AS-93-017, Sep 1993.
11. Hoang, Ngocdung; Howell, Steve; and Karangelen Nicholas, *Mission Critical System Development: Design Views and Their Integration,* NAVSWC TR 91-586, Naval Surface Warfare Center, Dahlgren, VA, Oct 1991.
12. Howell, Steve; Lock, Evan; and Nguyen, Cuong, "Integrating Systems Engineering Tools via an Interface Specification," *Computer Based Systems Engineering Workshop, Group D,* London, England, Dec 1992.
13. Howell, Steve; Lock, Evan; and Nguyen, Cuong, "Destination: An Environment for System Design Structuring and Allocation Optimization," *The Fifth Annual Software Technology Conference,* Apr 1993.
14. Howell, Steve and Nguyen, Cuong, *Systems Design Factors: The Essential Ingredients of System Design,* Naval Surface Warfare Center, Dahlgren, VA, NSWCDD/TR-92/268, Mar 1994.
15. Farr, W. and Smith, O., "A Tool for Software Reliability Modeling," *Proceedings of the SIMTECC'WNN 92,* 1992 International Simulation Technology Conference and Workshop on Neural Networks, Nov 1992, pp. 256-272.
16. Crigler, John R. and Chang, Michael Y., *A Statistical Characterization of the Effect of Tactical Loading on the AEGIS Tactical Computer System,* NSWCDD/TR-95/117, Naval Surface Warfare Center, Dahlgren, VA, Jun 1995.
17. Balci, Osman; DeVaux, David; and Nance, Richard, "Measurement and Evaluation of Complex Navy System Designs," *Proceedings of the 1993 Complex Systems Engineering Synthesis and Assessment Technology Workshop,* NSWCDD/MP-93/172, Naval Surface Warfare Center, Dahlgren, VA, Sep 1993, pp. 126-140.
18. Jones, Gary W.; Jons, Otto P.; and Ryan, J. Christopher, "Using Virtual Environments in the Design of Ships," *Naval Engineers Journal,* Vol. 106, No. 3, May 1994, pp. 91-106.
19. Ahrens, Judith and Prywes, Noah S., "Transition to a Legacy- and Reuse-Based Software Life Cycle," *Computer,* Vol. 28, No. 10, Oct 1995, pp. 27-36.
20. Crisp, Harry E. and Verma, Dinesh, "Second Annual Workshop on Engineering of Complex Systems in the 21st Century: Facing the Challenge," *Logistics Spectrum,* Vol. 29, No. 3, Fall 1995, pp. 70-73.
21. Crisp, Harry E. and Francis, James A., *First Annual Workshop on the Engineering of Systems in the 21st Century Proceedings,* NSWCDD/MP-95/79, Naval Surface Warfare Center, Dahlgren, VA, Apr 1995.
22. Crisp, Harry E. and Lee, Dona, *Second Annual Workshop on the Engineering of Systems in the 21st Century Proceedings,* NSWCDD/MP-95/189, Naval Surface Warfare Center, Dahlgren, VA, Oct 1995.

The Author

HARRY E. CRISP, II, is Program Manager for Engineering of Complex Systems, an exploratory development program sponsored by the Office of Naval Research. He graduated from Clemson University in 1964 with a B.S. degree in electrical engineering. He received an M.S. and a Ph.D. in electrical engineering from Auburn University in 1969 and 1971, respectively. He has been employed at NSWCDD since September 1971. He has performed research in digital control systems, served in the Mk 86 Program Office at the Naval Sea Systems Command, and has been the NSWCDD Program Manager for the Mk 86 Research and Development Program. He has been Head of the Weapons Control Technology Branch, the Information and Control Technology Branch, and the NSWCDD Technology Base Program Office. He has also served as director of the NSWCDD Independent Exploratory Development Program. He is a member of the Institute of Electrical and Electronics Engineers, the American Society of Naval Engineers, and the International Council on Systems Engineering.

High-Performance Distributed Computing for Surface Ship Combat Systems

Michael W. Masters

While the end of the Cold War minimized the threat of open-ocean naval combat, new littoral warfare missions and threats, such as tactical ballistic missiles, have emerged. These threats and missions challenge the traditional military standard computing resources available to the Navy—a challenge that is increasingly difficult to meet in a cost effective manner. Furthermore, in recognition of the high cost of military standards, Secretary of Defense William Perry has directed that commercial standards be utilized wherever possible. As a consequence of this changing environment, the AEGIS shipbuilding program has begun efforts to replace current military standard computing equipment with both higher performance commercial equipment and newer system architectures; i.e., distributed computing. As a part of this process, the AEGIS Program Office has collaborated with the Advanced Research Projects Agency (ARPA) to conduct a joint experiment on the feasibility of inserting a number of ARPA-developed distributed computing technologies into the AEGIS combat system. The High-Performance Distributed Computing Program (HiPer-D) was created from this joint initiative. The overall objective of HiPer-D is to validate distributed computing for Navy shipboard combat system applications. This paper presents the results of two major HiPer-D demonstrations of commercial computing technology as applied to the AEGIS combat system. It is shown that the HiPer-D prototype generally meets time-critical AEGIS antiair warfare (AAW) computing requirements using low-cost commercial off-the-shelf (COTS) equipment and software.

Introduction

Military computer technology was considered state-of-the-art throughout the 1960s and 1970s. The U.S. Navy's stringent requirements, specifically in its demands for real-time performance and simultaneous interface with a multitude of sensor and weapon systems, were consistently met only with custom-designed computer hardware and software. However, as Navy computing requirements have grown, the cost of continuing to meet those requirements with military standard computers has become prohibitive. At the same time, prices in the commercial sector have fallen—thanks in large measure to the advent of low-cost, high-performance microprocessor chips, inexpensive memory and disks, and local area networks. This trend has not gone unnoticed, and Secretary of Defense William Perry's initiative in 1994 to move toward commercial standards underscored the significant shift that was already underway in computing acquisition philosophy.[1]

Although the end of the Cold War minimized the threat of massive open-ocean naval combat, new littoral warfare missions and scenarios involving both combined arms and joint allied operations, have emerged. New threats, missions, and operating environments present a greater challenge to target identification, reaction time, command and control, and tracking and weapons accuracy, requiring even

more computer automation and data processing than in the past. As new systems designed to meet these challenges enter fleet service, it is increasingly difficult to engineer cost-effective computing systems with traditional military standard computing resources.

AEGIS System

The Navy's primary platform for delivering the new capabilities described above is AEGIS. The fleet of AEGIS ships comprises 27 CG-47 Ticonderoga-class guided missile cruisers and 6 DDG-51 Arleigh Burke-class guided missile destroyers, with another 29 presently under construction or planned. New warfighting capability developed to respond to fleet requirements is introduced into AEGIS ships through a well-disciplined combat system baseline process. A new baseline is implemented about every five years and is designed to provide a cost effective and low risk means of upgrading AEGIS combat capability. Presently, AEGIS Baseline 6 destroyers are under construction.

As Figure 1 shows, expected growth in AEGIS required computing capacity cannot be satisfied by scaling current military standard computer system resources. The current AEGIS computer suite has run out of capacity, and the architecture, based on Navy standard computers and architecturally limiting point-to-point interfaces, lacks the scalability needed to add the required new capacity. Figure 2 illustrates the use of the point-to-point architecture in AEGIS Baseline 5. A cost-effective long-term architecture, capable of being expanded as requirements grow, is needed.

AEGIS Transition to Commercial Computers

As an intermediate step, AEGIS is planning limited use of commercial computers as "adjunct processors" to augment the Navy standard computers to meet Baseline 6 computing requirements. However, by Baseline 7, which will be authorized in fiscal year 2000, the additional computing capacity made available by adjunct processors will be exceeded (if not rendered unaffordable). A solution based on full use of COTS computing resources is needed. Planning for that transition has already begun and, in fact, AEGIS is currently performing studies to determine whether or not the AEGIS Weapon System can be transitioned to COTS technology by Baseline 6 Phase 2A—an FY98 ship. Figure 3 illustrates a notional AEGIS combat system based on commercial computing and network technology.

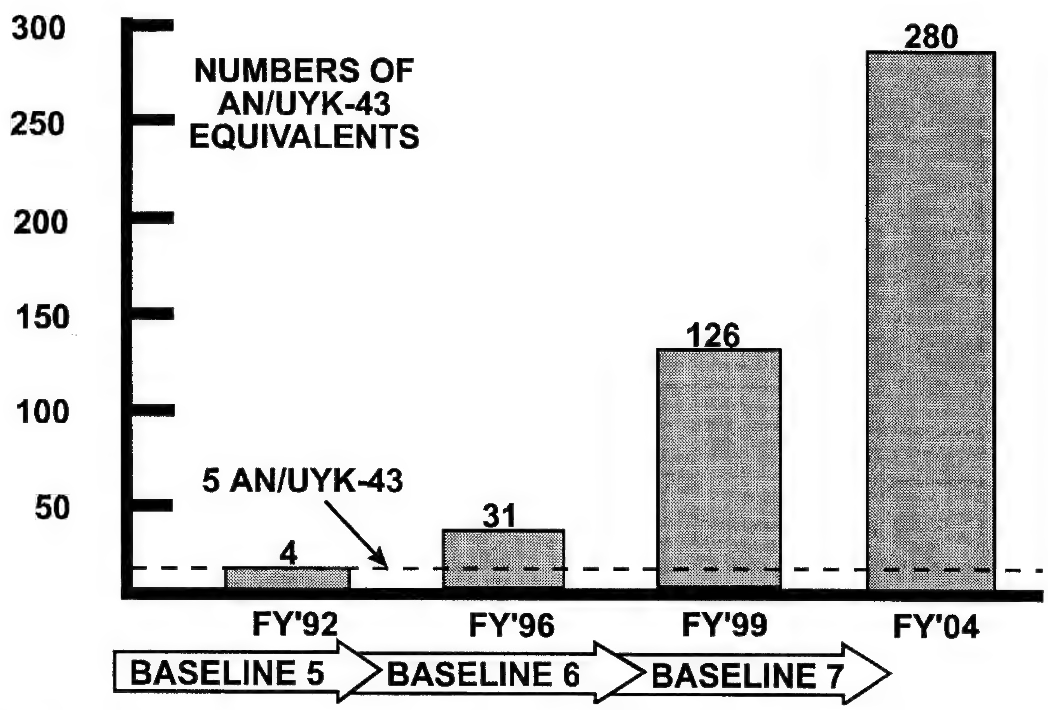


Figure 1. *Operational requirements growth mandates AEGIS Combat System computer system upgrade.*

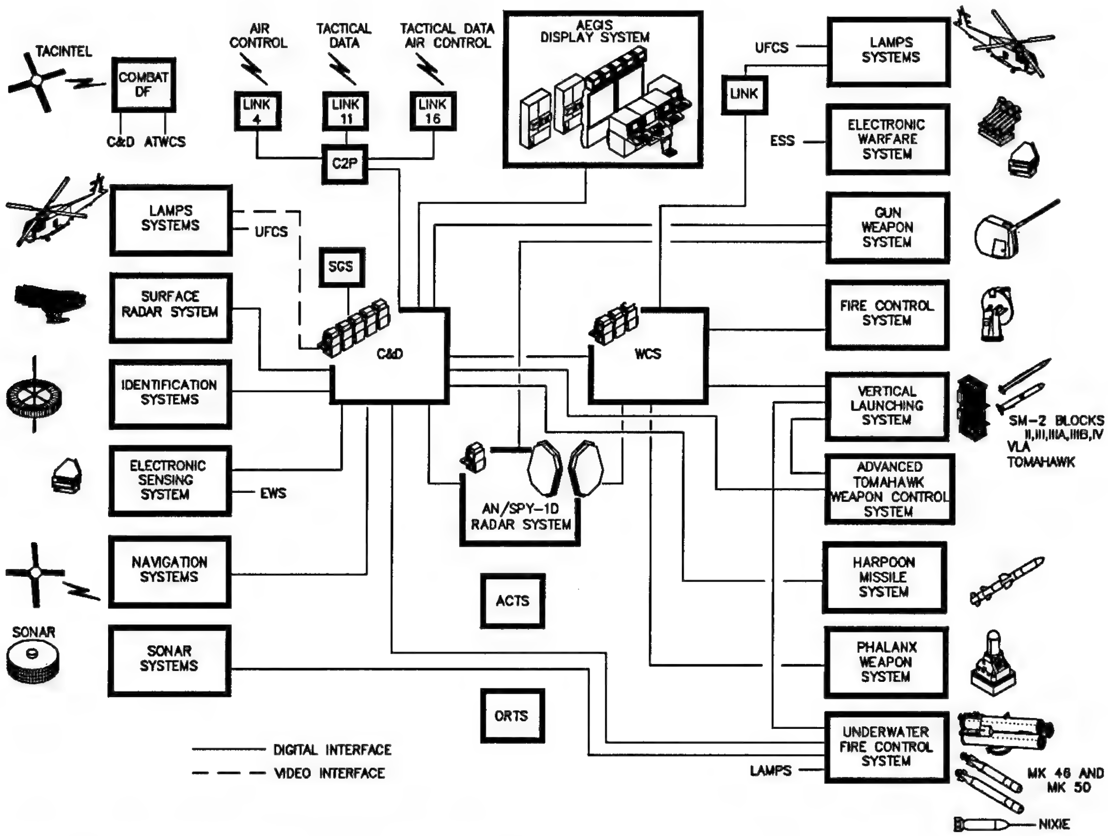


Figure 2. *The AEGIS Baseline 5 Configuration is based on Navy standard computers and Navy Tactical Data System (NTDS) point-to-point channels.*

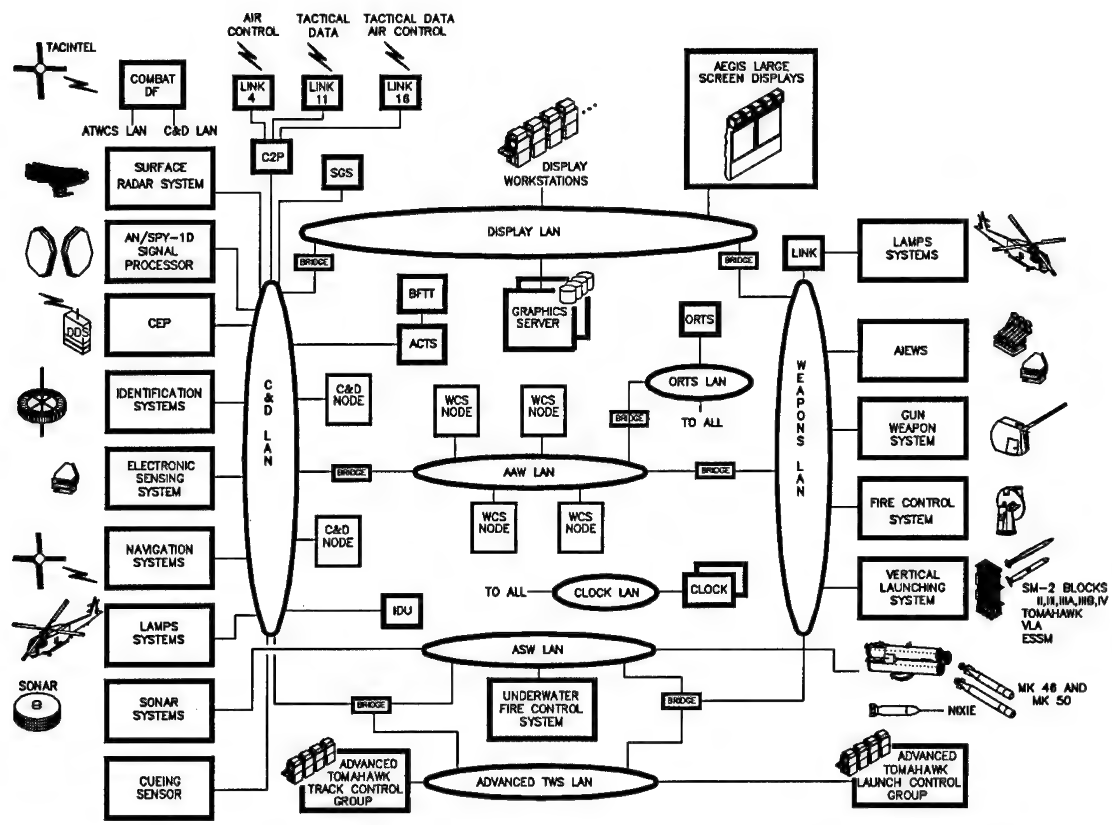


Figure 3. *A notional future AEGIS baseline based on COTS computing technology.*

However, transition to commercial computing technology will not be as painless as individuals unacquainted with the technical requirements of Navy weapons control have assumed. While it is not well appreciated outside the Navy weapons control community, the requirements of sensor systems, control systems, and weapons systems impose real-time performance requirements that cannot be solved simply by upgrading processor speed or network "bandwidth" (the total amount of data that can be transferred through a network in a given time). While speed is admittedly important, time-critical military applications also require predictability and precise control over the flow of processing. This means that designers must have means of explicitly controlling the following parameters:

- How much time operations will take
- That operations will not occasionally run on indefinitely
- That operations can be made to happen exactly when needed

Computer Needs—Commercial versus Military

In this regard, the commercial marketplace has not been nearly so demanding of the computer industry as the military environment. For example, it may be acceptable to the user of an automatic teller machine (ATM) to wait if the computer network serving the ATM is busier than usual. In the commercial aviation community, it is acceptable for planes to circle the airport in a holding pattern if the number of planes landing at an airport temporarily exceeds the capacity of the local air traffic control system. However, one cannot expect an incoming hostile missile to pause until an overloaded fire control computer can react to it. These scenarios are presented only to contrast the differences in the market forces driving the commercial computer industry versus the threat-based requirements of military applications.

Another area where AEGIS requirements exceed those of most commercial users is that of communication between computers and other devices. In order to guarantee service, military computers have generally used dedicated connections called channels, a solution once widely used in commercial computing as well. However, as the number of computers and other devices to which computers interface increases, the number of channels required becomes prohibitively expensive. Some form of shared network is clearly required for intercomputer communication in the future. However, many commercial networks, including "Ethernet," which is commonly used in office Local Area Networks (LAN), do not provide the system designer with a way of ensuring that transmission deadlines can be met. Ethernet, for example, resolves conflicting information transfer requests by making each requesting computer delay its transmission of data for a random amount of time and then trying again. Again, a hostile incoming missile will not entertain a request to wait and try again later when the defensive missile launcher is free to respond.

This is not to say there are not suitable products available in the commercial computer marketplace. For example, newer network technologies now available do provide the level of control needed in the example in the preceding paragraph. Additionally, commercial operating systems are beginning to provide the precise control features required for real-time computing. The challenge facing AEGIS is that the number of suitable products is only a small percentage of the total computer marketplace. Thus, carefully engineered solutions are required to solve AEGIS computing problems.

The efficacy of proposed solutions must be validated with critical system engineering experiments and hard engineering data. This process involves evaluation and selection of COTS products in concert with a new system architecture—an overall system design—that focuses in on the critical time requirements of the AEGIS combat mission. Furthermore, mechanisms are needed to positively influence the computer industry to provide COTS products that meet emergent AEGIS needs. Given the large size of the nonmilitary marketplace, this can best be accomplished by gathering engineering data that demonstrates the deficiencies of commercial computing products when used for real-time purposes and by participating in commercial standard-setting organizations.

The HiPer-D Program

It was for these reasons that HiPer-D was created. HiPer-D is a critical experiment using new computer technology conducted jointly by the Department of Defense's ARPA and by AEGIS. The purpose of the experiment is twofold:

1. To evaluate the suitability for Navy use of new computing technologies developed by ARPA
2. To address technical issues associated with transitioning AEGIS ship combat systems to modern commercial computing technology

The AEGIS program has been actively engaged in exploring the use of new combat system architectures and commercial computing technologies for a number of years. This involvement was greatly accelerated when, in early 1991, an agreement was reached between the ARPA Computer System Technology Office and the AEGIS Shipbuilding Program Office to conduct a joint experiment on the feasibility of inserting a number of ARPA-developed distributed computing technologies into an AEGIS combat system application. ARPA has been engaged since the early 1970s in the development of high-performance computing elements, such as parallel processors, distributed computing, portable secure operating systems, and high-speed networks. ARPA offered three technology products for evaluation by AEGIS:

1. The Intel Paragon parallel commercial supercomputer
2. The Mach operating system, known commercially under the name of OSF1
3. The Isis distributed processing toolkit

The HiPer-D Program commenced in June 1991 with an overall goal of conducting a critical experiment to assist AEGIS in making the transition from a federated Navy standard computer architecture to a commercially-based distributed architecture. Structured as a six-year program, the HiPer-D financial plan called for ARPA funding of the first three years and AEGIS funding of the next three years. An Executive Panel was formed consisting of ARPA and AEGIS program managers, and leading program personnel from the three technical organizations were chosen to perform the engineering work of the program:

- Johns Hopkins University/Applied Physics Laboratory (JHU/APL)
- Martin Marietta (AEGIS prime contractor)
- Naval Surface Warfare Center, Dahlgren Division (NSWCDD)

Under the Executive Panel, the Technical Management Team was formed to manage day-to-day operations and coordinate with ARPA's university and industry technology providers. These included Carnegie-Mellon University, Cornell University, the Open Software Foundation (OSF), and Intel Corporation. The program was implemented by the three technical organizations.

HiPer-D Thrusts

The HiPer-D Program was divided into three technical thrusts: (1) system engineering, (2) technology demonstration and evaluation, and (3) technology development and feedback. Lessons learned are fed back to ARPA's academic and industry developers and to the computing standards community. Figure 4 illustrates the relationship between these initiatives.

Thrust One examined several computer system architectures and evaluated each with respect to a set of metrics that spanned not only traditional computing efficiency measures, but engineering practices and cost effectiveness as well.[2] The architecture shown in Figure 3 is taken from the study. A derivative of that architecture is the primary design being considered for implementation in Baseline 6 Phase 2A.

Thrust Two involved the development of a realistic, large-scale benchmark demonstration that could be used for two purposes. The first was to provide a testbed through which the latest ARPA and other computing technology flows. This provides the AEGIS community with early access to the latest technology, and it provides ARPA with early user feedback from an engineering environment, greatly reducing the time required to transition technology. The second purpose was to develop and evaluate new, open and distributed-computing design techniques. This second goal was important because, when AEGIS abandons Navy standard computers, it will also leave behind its existing base of operating systems, programming languages, support software, display software,

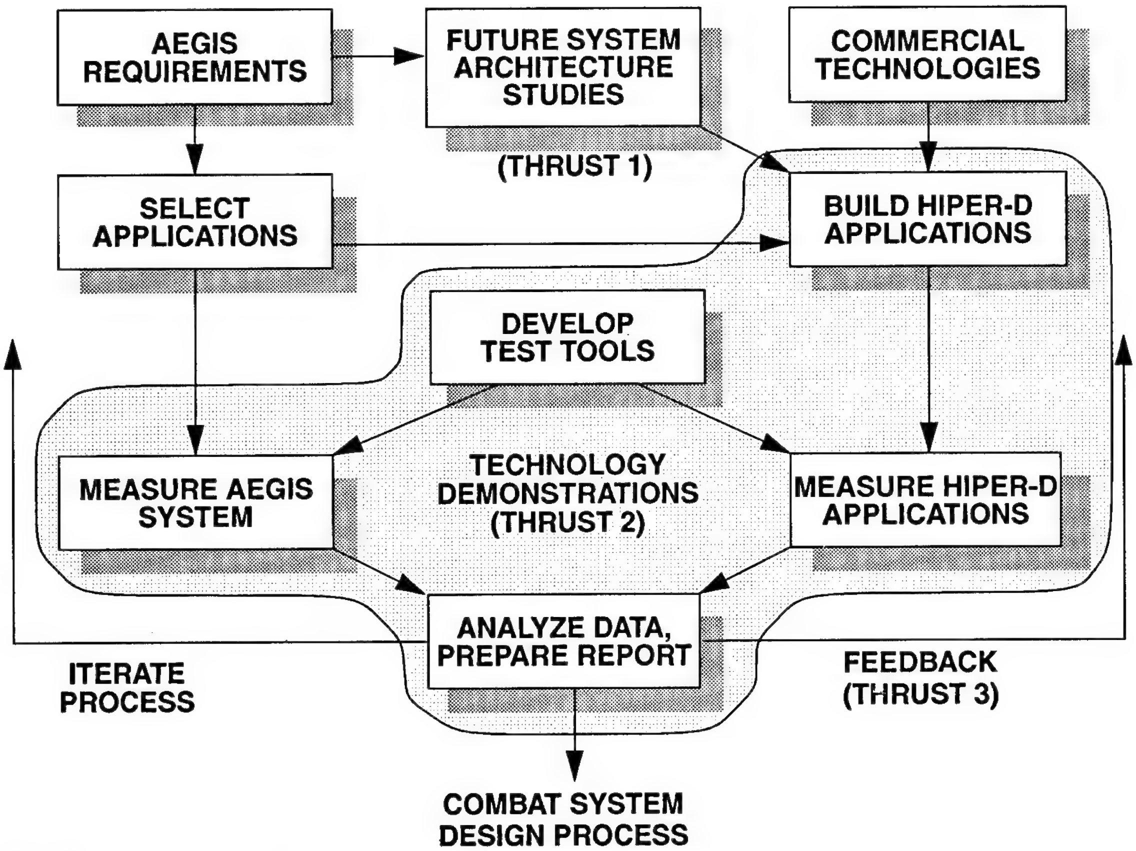


Figure 4. *HiPer-D technology experiment derives from both AEGIS requirements and emerging commercial computing technology.*

communication protocols, etc. Shifting to commercial computers is not just a matter of running old computer programs on new equipment. A new computer system and software architecture will also be required—a change of unprecedented scope.

Thrust Three was included in the program to ensure that the lessons learned from the experiment would be available for incorporation into future technology products. This feedback is important because much of the commercial computing marketplace is driven by market influences unrelated to real-time requirements. Without Navy feedback, especially in the form of quantitative requirements and measured engineering data, it is possible that critical military capabilities and performance characteristics will not be present in many COTS products.

The Thrust Two Experiment

Although the HiPer-D Program is a complex and multifaceted program, this article concentrates on the Thrust Two experiment. Two major milestones have so far been reached. The first, funded by ARPA as a part of HiPer-D's Phase 1, was Integrated Demonstration One (I1), conducted in March 1994. I1 employed the three commercial technologies supplied by ARPA. As HiPer-D entered Phase 2, the test nomenclature was changed to reflect a stronger AEGIS baseline orientation. The first milestone under Phase 2 was Engineering Test One (T1), conducted in May 1995. T1 used updated versions of Isis and OSF1 along with the Fiber Data Distribution Interface (FDDI) network technology and a large number of commercial workstations. The Intel Paragon was not used in T1 due to real-time input/output problems discovered during I1 that had not been overcome as of the date of T1. Both tests consisted of an evaluation of a prototype of certain time-critical combat system functions that spanned the AEGIS AAW capability from target detection to missile engagement order. This AAW "path" provided a through-the-system test of the ability of COTS technology

to meet AEGIS mission-critical computing requirements.

The discussion that follows first focuses on the design concepts that were identified for evaluation as a part of the program. Then it describes the tactical applications that were chosen to implement the time-critical Standard Missile 2 "AAW path" through the combat system. Finally, this article describes the I1 and T1 experiments.

Overall, the results of the I1 experiment were mixed. Considerable progress was made in dealing with distributed processing issues such as capacity, scalability, fault tolerance, and open system design. However, I1 also demonstrated that much remains to be done in the area of operating system predictability and intercomputer communication effectiveness. Many of the problems surfaced during I1 were addressed the following year, and their solutions were incorporated into T1. Though data analysis and assessment for T1 is still ongoing, it now appears that the HiPer-D COTS-based prototype is able to meet at least some of the AEGIS time-critical AAW Standard Missile 2 engagement requirements, particularly those in the all-important area of "auto special" popup air target engagements.

Open and Distributed-Computing System Designs

It was decided early in the program that new design concepts and a new system architecture, appropriate to a full transition to commercial computing technology, would be aggressively pursued. It was also decided that the limitations experienced by AEGIS in the area of computing capacity and scalability dictated a focus on the issues of flexibility for future change and cost-effective, long-term maintenance. Reimplementing old designs on new equipment without a new architecture would gain little, if anything, in flexibility of design. A number of key characteristics were chosen as defining attributes for the new architecture. These technology attributes constituted the primary focus of this investigation.

- COTS equipment
- Distributed-computing architecture
- Flexible, open-system design
- Adherence to standards
- Continuous availability through fault tolerance
- Scalability

Distributed computing is architecturally different from centralized computing. It involves partitioning the computing tasks of a large-scale system into many small processes, or computer programs. These many small programs can then be allocated to a large number of computers, thus taking advantage of the combined processing power of all of the computers. The available computers are usually interconnected by a shared network of some type since the sheer number of computers makes point-to-point connections prohibitively expensive.

Just as distributed processing is the key to solving problems of capacity and scalability, open-system design is the key to achieving ease of change and maintenance. Informally, an open system is a system for which change and growth of both functionality and capacity can be accomplished with minimum cost and impact on existing system components through the use of widely accepted hardware and system-software standards and standard application components and well-defined interfaces. Open designs encompass two major aspects: nonproprietary commercial standards and application design conventions. Historically, Navy systems have not used open designs. One probable explanation is that the software layering needed to build open systems makes it difficult and expensive to meet real-time performance objectives in Navy standard computers. However, the availability of affordable, modern high-performance computers bodes well for the viability of open-system principles in the future.

Commercial Standards and a Multilayered Approach

Consistent with Secretary of Defense William Perry's direction, the Navy intends to use commercial standards to the maximum extent possible in moving toward COTS. Standards foster interoperability of equipment, computer programs, and systems. Standards reduce the cost impact of proprietary equipment on system development. Use of standards does not imply standard equipment. Rather, it implies widely

accepted specifications by which a competitive range of vendors can produce components for use in systems in an interchangeable manner. In the critical area of software development, a standard programming interface is provided for application developers, thus reducing development, maintenance, and programmer training costs.

While the role of standards for hardware and system software in supporting open systems is widely recognized, the role of application design is not yet so well understood. Nevertheless, application design plays a critical role, especially in large-scale systems where sheer size and longevity provide incentives for systemwide, life-cycle cost optimizations. In the commercial world, the best engineering practices tend to produce designs that are highly layered for flexibility and separation of concerns (functional partitioning). Such designs appear to be robust and cost effective in the face of growth and change. A good example is the International Standards Organization's communication protocol stack reference model—a seven-layer abstraction that spans the entire range of communication services from application to wire.

An analogous multilayer approach is being pursued in HiPer-D. Figure 5 illustrates this concept. Evaluation of commercial system software products at the lowest layer was a fundamental goal of the HiPer-D critical experiment. At the next level up—the distributed system infrastructure layer—a number of design efforts were initiated both within HiPer-D and in industry and academia. These design efforts were intended to address the areas of distributed system control, fault tolerance, support for scalability through parallel design techniques, and system resource management. At the common tactical support layer, open, scalable design techniques were developed for track file distribution, display, and data extraction. Redesign at the application level was also examined. Distributed system instrumentation, testing, and debugging were additional areas of focus in the ongoing experimentation and technology evaluation.

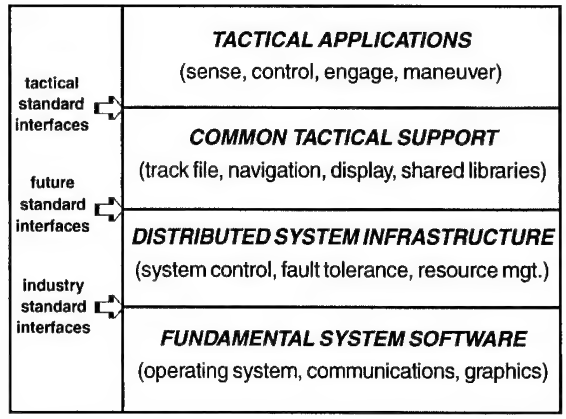


Figure 5. *The open-system approach uses standard interfaces to communicate between layers of software.*

Application Computer Programs

The fundamental operational paradigm of the AEGIS Combat System can be captured in three words: detect, control, and engage. The entire combat system—encompassing a multitude of sensors, weapons, and computers—was engineered to perform these key naval warfare operations. More than any other surface combatant in the history of the Navy, AEGIS was engineered with end-to-end performance as the goal. Accordingly, it was decided that any assessment of new computer technology for AEGIS must address the problem of performance throughout the detect-control-engage process.

Performance requirements are nowhere more stressing nor demanding than in the AAW domain. As a result, three time-critical tactical application functions were chosen to represent the AAW "target detection to missiles away" path through AEGIS. These three functions were:

1. Radar contact correlation and target tracking
2. Target identification
3. Threat evaluation, weapon assignment, and engagement management

Each of these functions was designated to be prototyped by one of the participating technical organizations. Target parameters, radar hardware, missile flyout, and terminal homing were all simulated. Figure 6 illustrates these functions and the flow of information among them in a combat system.

The HiPer-D Correlator Tracker and the Air Engagement Controller. Figure 7 is a functional block diagram of the 11 prototyped computer programs. The radar front end, referred to as the HiPer-D Correlator-Tracker, incorporated several functions from a new tactical system initially planned for backfit on AEGIS cruisers as well as other non-AEGIS ships. Designated the Cooperative Engagement Capability, this new system will

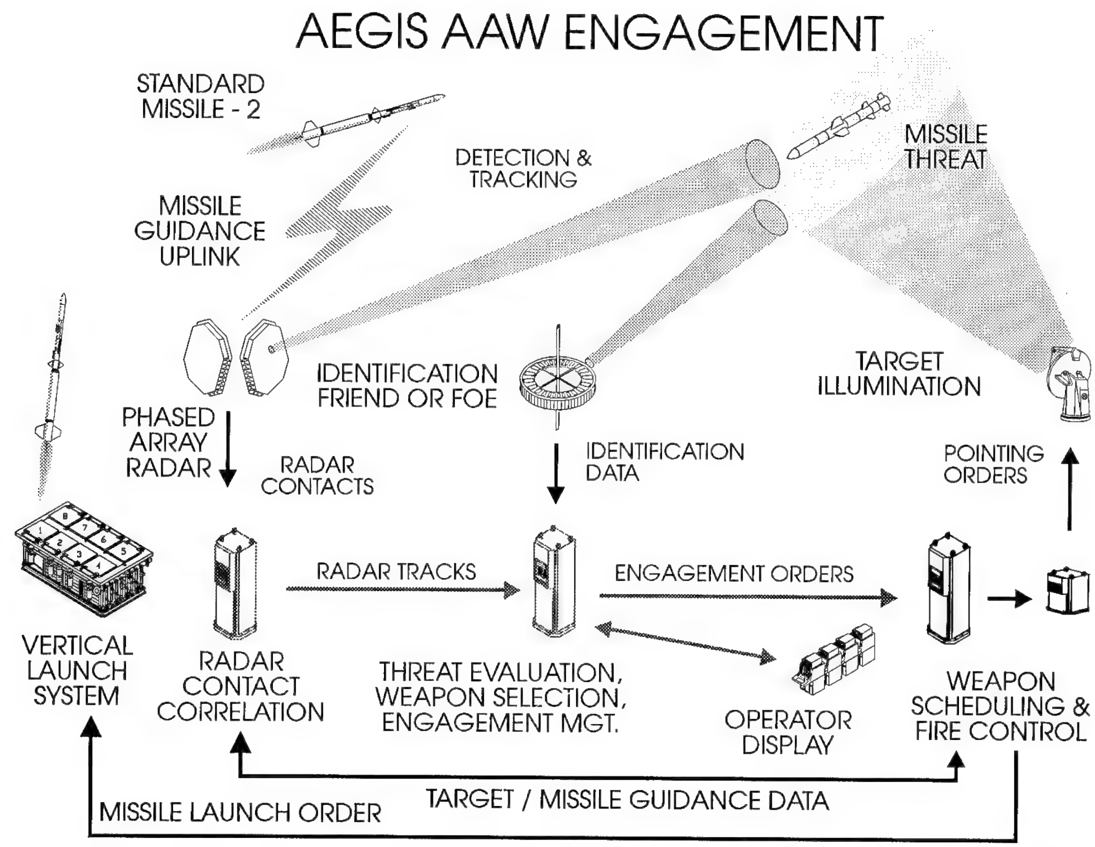


Figure 6. *The AEGIS antiair engagement mission places stringent real-time performance requirements on the computer system.*

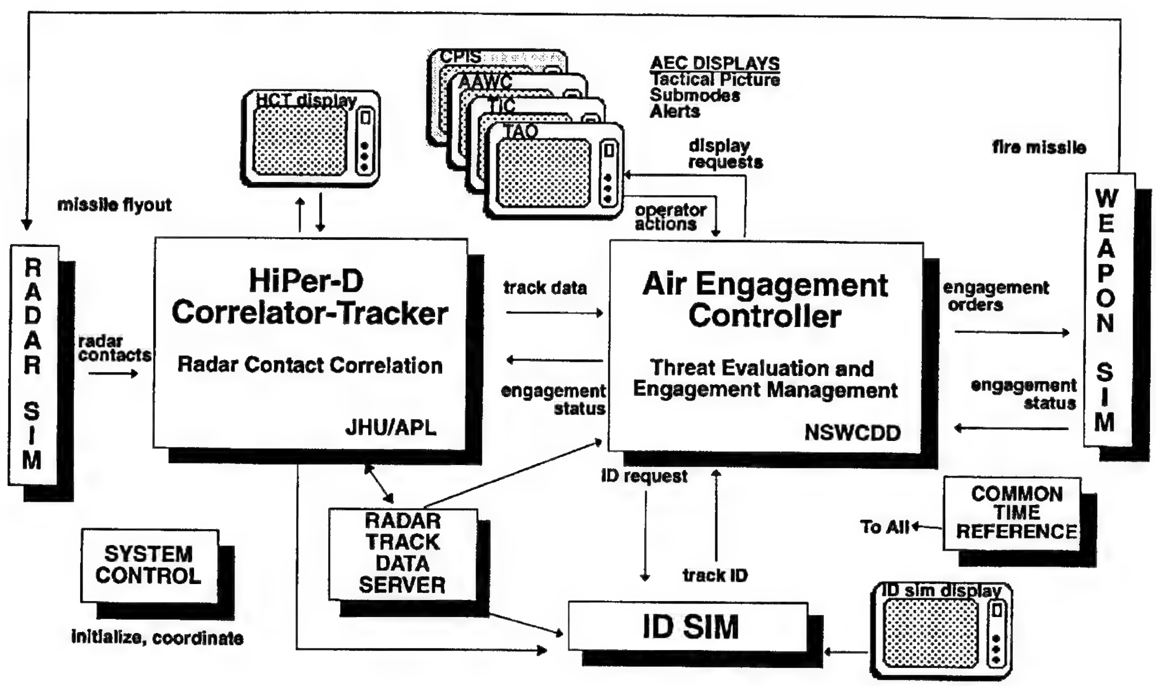


Figure 7. *Integrated Demonstration One implemented key segments of the AEGIS antiair engagement path through the combat system.*

permit suitably equipped surface combatants to share radar data, thus enhancing and integrating fleet knowledge of a tactical environment. The target identification component was to be an adaptation of a future AEGIS ship upgrade addressing identification deficiencies in the joint littoral operational environment. This component could not be completed in time for I1; however, the identification function was simulated. Finally, selected portions of the threat evaluation, weapon selection, engagement management, and display functions of the AEGIS Command and Decision System were reprogrammed in Ada. This application was called the Air Engagement Controller.

Both of these applications consisted of several computer programs. The HiPer-D Correlator-Tracker and its simulators now comprise eight unique executable programs, with approximately 150,000 lines of code in the C programming language. The Air Engagement Controller and its simulators consist of ten distinct, executable programs, with approximately 125,000 lines of code in the Ada programming language. Continuous availability was achieved by replicating several of the programs for fault tolerance. Multiple copies of a fault tolerant program were run simultaneously in different computers, each receiving the data it needed to perform its function. One copy was designated as the primary; any other copies were shadow versions. If a primary program copy failed, for example, because its computer had become a casualty, then the shadow copies detected the loss of the primary, with one copy taking over the processing of the failed primary.

Radar Track Data Server (RTDS). In keeping with the goal to aggressively examine new design techniques that fully exploit commercial computer technology, a number of design innovations were incorporated into the prototypes for evaluation. Foremost among these was use of the commercially popular "client-server" design model. In this model, service consumers (clients) interface with service providers (servers), according to an open standard interface.

One of these servers was the RTDS. The purpose of this component was to provide simulated radar track data to users on demand. It was composed of a number of replicated copies designed to provide both fault tolerance and load sharing by means of replication. For I1, several radar data consumer clients in the demonstration system were replicated for fault tolerance but not for scalability.. For T1, one of the radar data consumer programs, the "auto SM"—or automatic Standard Missile 2 (SM-2) doctrine program—was redesigned as a scalable peer client of the RTDS. This program automatically compares air track kinematic and geometric characteristics against predetermined threat profiles. This process uses automated "if-then" rules, called doctrine statements, that are used by AEGIS to identify situations where a combat system response is advisable or automatically implemented.

Integrated Demonstration One

The previously described software components were assembled and integrated onto a hardware test-bed composed of an Intel Paragon computer with the Mach operating system and a number of commercial workstations running the commonly available UNIX operating system. The Paragon contained a total of 23 processors. The workstations and the Paragon communicated via Ethernet, the only standard commercial LAN available for the Paragon at the time of the test. The Isis distributed-processing toolkit was used for most intercomputer communication. Including replicated copies of programs in the count, there were 17 programs on the Paragon representing prototypical AEGIS functions. There were also 22 support programs resident, providing startup control, time synchronization, data extraction, and performance monitoring. Approximately 20 additional tactical and support programs resided on the workstations. Figure 8 shows the equipment configuration and the allocation of these programs to the equipment.

Measurement and assessment methods and tools were developed for the experiment. This involved instrumentation of not only the demonstration system but also the AEGIS Weapon System itself. During the integrated demonstration time period, measurements were taken on the AEGIS Baseline 4 Phase 2 system in the AEGIS Computer Center at NSWCDD in Dahlgren, Virginia. These measurements were laid out in a timeline representing the SM-2 engagement sequence operating under control of automatic standard missile doctrine. Under automatic

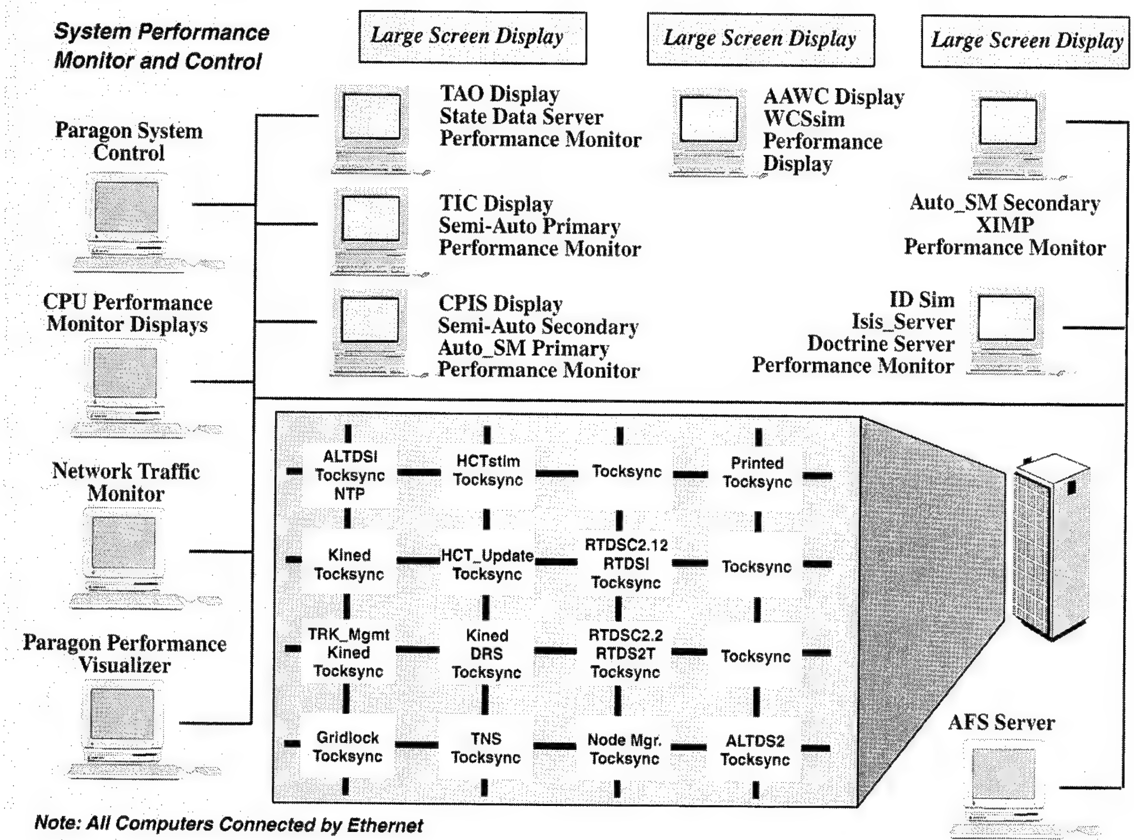


Figure 8. *Integrated Demonstration One contained commercial parallel processor, network, and workstation technologies.*

doctrine, targets may be engaged automatically without operator intervention. This mode of operation was chosen for measurement and evaluation, because it removed operator variability from the measurement process. A comparable sequence and timeline was constructed based on performance measurements using HiPer-D Program software representing analogous functionality. Figure 9 illustrates these timelines for comparative purposes.

I1 Overall Results. The formal I1 took place in March 1994. Data collection and analysis continued over the following two months. The data analysis produced a great deal of valuable information about not only the COTS HiPer-D implementation but also the AEGIS system. The data from AEGIS is classified and cannot be presented here. However, a number of quantitative comparisons and conclusions can be drawn. A detailed description of the I1 test configuration, the test results, and the lessons learned is contained in Reference 3. Overall qualitative results were first widely reported to the Navy community in Reference 4.

The products evaluated, while obviously moving toward a real-time capability, were not able to reach AEGIS-required performance levels in I1. Perhaps the most striking result was that while the products used delivered a great deal of raw performance, they did not provide the predictable, bounded control over computing operations that the Navy requires. "Best case" results for timed deadline scheduling were equivalent to results observed in Navy standard computers, in the range of 0 to 5 ms of the scheduled wake-up time under loads. UNIX-based commercial workstations performed similarly. Given the availability of a preemptible real-time operating system, these computers should eventually be able to meet timed, periodic requirements. However, best case application-to-application network

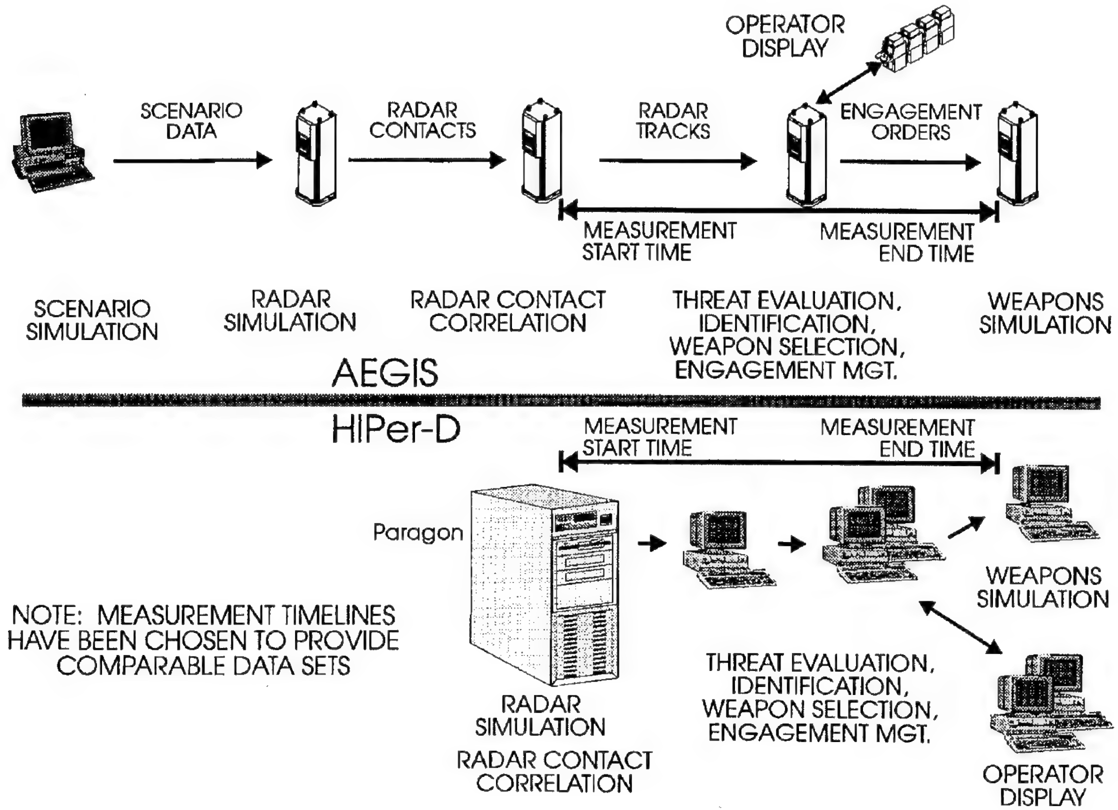


Figure 9. *HiPer-D's Integrated Demonstration One compared the performance of key segments of the AEGIS antiair engagement path with similar functions built on commercial computing technology.*

message-passing times were about 15 ms, which is slower than dedicated Navy Tactical Data System (NTDS) point-to-point channels by two orders of magnitude. These results are consistent with other network benchmarks in heavily workloaded computer systems.

Problems Encountered during I1. Worst case values of a few hundred milliseconds were occasionally observed for missed deadlines, and over a second for message-passing times. These results were attributable to two factors. The first was lack of operating system preemption—the ability of the operating system to recognize and temporarily suspend low priority tasks for high priority tasks. The second was inefficiency in Paragon communication to and from the commercial workstations. This proved to be a major bottleneck. During periods of peak scenario activity and during failure detection and recovery events, delays of up to several seconds were intermittently observed in getting message traffic out of the Paragon.

Detailed evaluation revealed that the problem lay with an inefficient external communication structure to and from the Paragon that severely impacted the ability of Isis to perform its function. Isis was designed based on a nominal cost on the order of 50 ms to perform routine network communication services such as socket reads. Standalone tests on the Paragon revealed values greater than 5 ms—a difference of over two orders of magnitude. The reasons behind this disparity were threefold:

1. Inefficiencies in Mach, including the lack of kernel preemption, a critical real-time operating system feature
2. An unsophisticated implementation of Ethernet communication protocols
3. The Paragon requirement that all Ethernet message traffic go though a "service" processor devoted to external Ethernet communication

A major problem with Isis was failure detection. In its current form, failure detection was too slow to provide the continuous availability attribute that AEGIS requires. Although software process crashes were detected in the best cases within a few tens of

milliseconds, hardware failure detection settings for Isis could not be safely set below about 10 seconds. By contrast, combined failure detection and recovery requirements for AEGIS computer failures are on the order of one second.

Overall, best case results showed that the commercial components used in I1 provided performance close to AEGIS requirements for the portion of the AEGIS engagement path measured. However, this level of performance could not be achieved reliably. Worst case values were several hundred percent above AEGIS requirements. The greatest barrier was lack of operating system scheduling determinism and lack of low latency communication. (Latency is the delay a message encounters between the time the sending application requests that it be sent and the time the receiving application actually has it.) However, despite the limitations of the first experiment, good progress was made toward functional use of parallelism, via replication of programs, as a means of achieving fault tolerance and scalability. This highlighted the need for operating systems and communication services designed with real-time requirements as a goal.

Engineering Test One

As a result of the deficiencies found during I1, a number of changes were made for the T1 test. To begin with, the Paragon was eliminated from the configuration. The assessment was that the input/output difficulties it experienced during I1 could not be overcome in time for T1. Commercial workstations were substituted. ARPA support was gained for a continued relationship with the OSF in order to secure use of OSF's experimental real-time version of the OSF1 operating system, a version called OSF1-MK7. Since OSF requires initial development on personal computers, a small number of personal computers were added to the test configuration for T1. The Isis communication toolkit was retained for T1.

FDDI Network and Other Enhancements. In the networking area, an FDDI network was inserted into the configuration for T1. This decision proved invaluable. Data collected during the T1 test revealed that peak network loading during the test reached approximately 30 megabits per second, far in excess of the practical Ethernet limit of six or seven megabits per second—but well below FDDI's theoretical maximum of 100 megabits per second. Furthermore, a new FDDI topology, called Survivable FDDI with Concentrator Tree, provided network hardware fault tolerance for the first time in a HiPer-D test. This NSWCDD-developed design is under consideration for use in AEGIS Baseline 6. Figure 10 illustrates the T1 configuration.

The AAW prototype was greatly enhanced for T1. Track capacity was targeted to increase from an entirely inadequate level of 100 tracks at I1 to 1000 tracks for T1. This decision proved to be extremely fortuitous. The order-of-magnitude increase in capacity became a major design driver for T1, and it forced the HiPer-D team to seriously address capacity as a critical system metric. In addition to increasing track capacity, the AEGIS auto-special engagement capability was added to the prototype. Auto-special is the most time stringent AAW requirement in AEGIS. Further, scalable auto Standard Missile (SM) doctrine clients were added as a complement to the scalable RTDS, thus providing a complete example of scalable processing for both clients and servers. Finally, tactical code from the AEGIS SPY-1 phased-array radar—about 100,000 lines of Ada code converted from CMS-2—was incorporated into the configuration. The SPY-1 code currently provides internal Dynamic Test Targets. Figure 11 contains the functional block diagram of T1.

Instrumentation Improvements: Jewel and Event Trace and Analysis Package (ETAP). Instrumentation was a major focus for T1. A shareware instrumentation tool, Jewel, was obtained from Germany and adapted for use in HiPer-D. Jewel has components that run on each computer under test. Time tagged data is extracted into a shared memory area on each computer and is periodically transferred across a network interface to a collection computer, which records the data and makes it available for visualization. One of the major advances HiPer-D has contributed to large-scale system development is the ability to observe critical internal program parameters during a test.

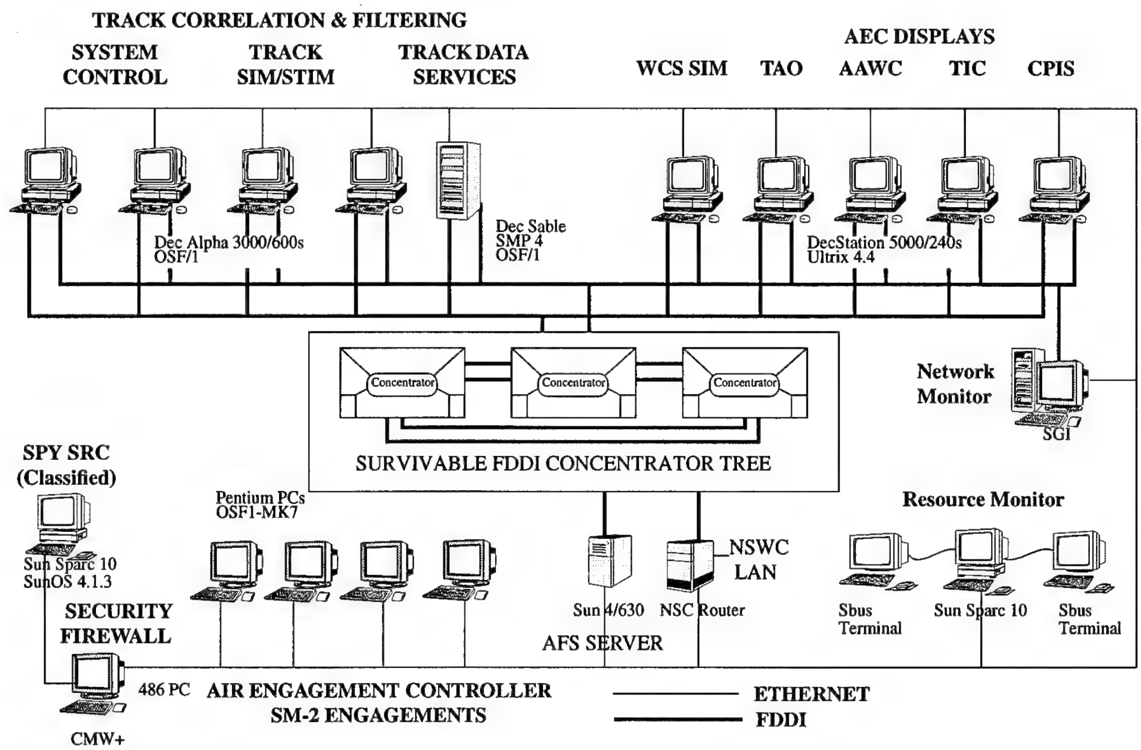


Figure 10. *Engineering Test One used commercially available computers, FDDI network technology, and the Open Software Foundation's experimental real-time operating system.*

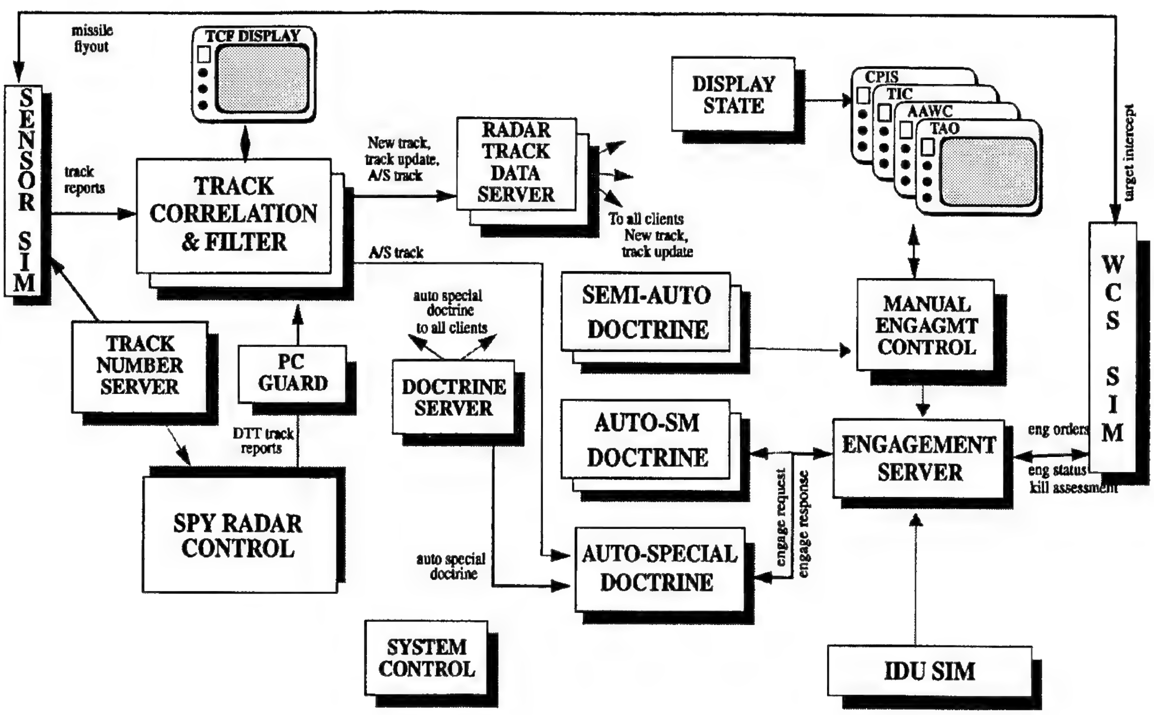


Figure 11. *Engineering Test One added the time-critical AEGIS auto-special Standard Missile 2 engagement capability to HiPer-D's AAW prototype.*

Figure 12 illustrates the HiPer-D instrumentation approach.

In addition to Jewel, HiPer-D worked with OSF to create an instrumented version of the OSF1-MK7 kernel. This kernel instrumentation, called the Event Trace and Analysis Package (ETAP), has provided essential insights into application program performance. For instance, it played a vital role in allowing the HiPer-D prototype to be tuned in order to reach its goal of processing 1000 tracks for T1. ETAP has been provided to the Portable Operating System Interface (POSIX) community for evaluation in the development of a future POSIX standard for operating system instrumentation.

Goals Met in T1 Results. The results of T1 are very exciting. Data analysis and evaluation is not complete at this point. Some sources of nondeterminism remain, notably the Isis distributed processing toolkit and the non-real-time operating systems employed on some HiPer-D computers. However, preliminary results indicate that many of the goals of T1 have been achieved. The goal of processing 1000 tracks was reached. More importantly, the critical auto-special AAW engagement timeline has been met. Several factors were crucial in reaching these goals. First, the highest performance workstations in the configuration, those based on the latest processor technology available today, appear to be capable of providing ample processing power for the AAW timeline. Second, the use of FDDI network technology, as noted earlier, provided an essential margin of high bandwidth communication for the HiPer-D prototype. Third, the use of the OSF1-MK7 real-time operating system, with its preemptive kernel, in the critical auto-special engagement path, permitted the HiPer-D prototype to be tuned to meet the timeline requirements of the AEGIS AAW engagement capability.

Added to this is the fact that the HiPer-D prototype is based on high payoff commercial design concepts such as open client-server designs, replication for fault tolerance and scalability, and a layered software infrastructure of shared reusable libraries. These design concepts promise to make software and system construction and maintenance both faster and cheaper in the future. The fact that the HiPer-D prototype generally meets time-critical AEGIS AAW engagement requirements while being based

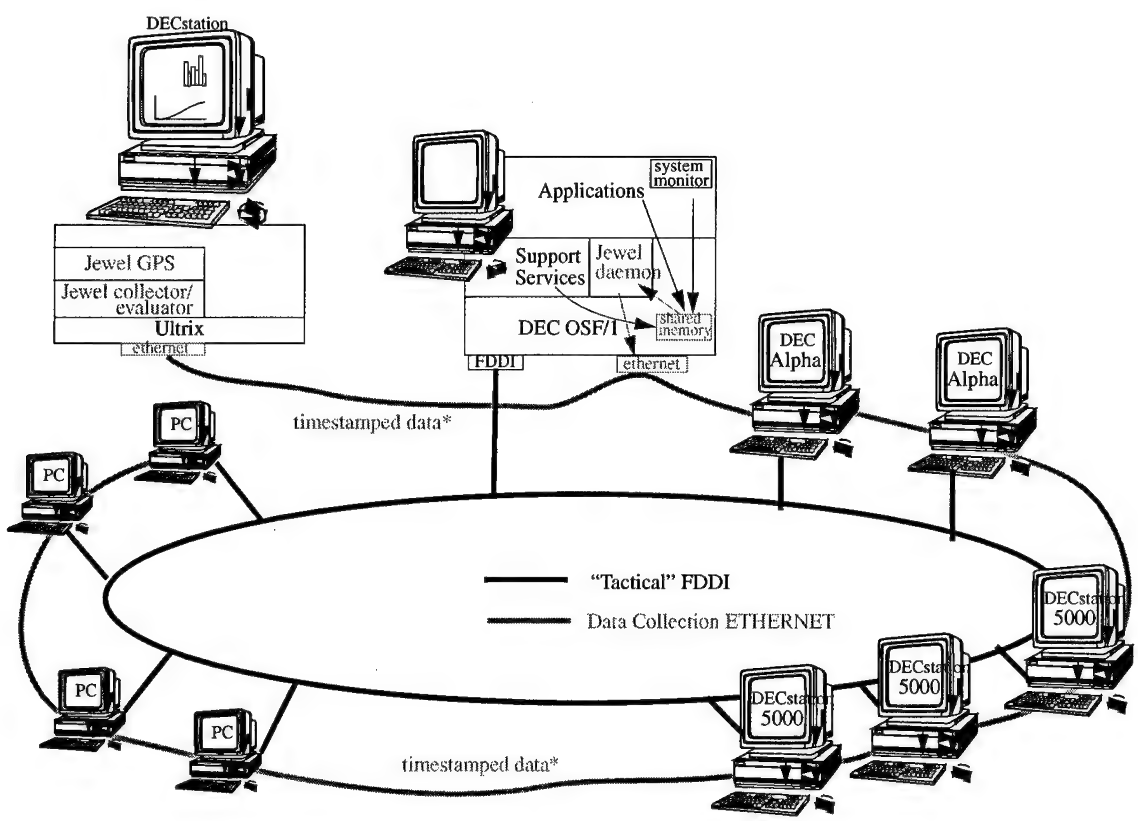


Figure 12. *Instrumentation was a major focus of HiPer-D's Engineering Test One.*

on low-cost COTS equipment and system software and on cost-containing open-system design principles should be considered a major achievement in the evolution of Navy combat systems.

In recognition of the value of the HiPer-D instrumentation, the AEGIS prime contractor, Lockheed Martin Corporation (formerly Martin Marietta), has requested its delivery plans for use in the development of Baseline 6 Phase 2A. This transition is an important milestone in HiPer-D's efforts to develop a system test process for COTS-based distributed systems.

Future Work

For ARPA, HiPer-D has been successful in providing the kind of feedback that was sought going into the program. For AEGIS, a method has been established to ensure that the risks associated with use of commercial computer technology can be assessed and mitigated prior to commitment to production. Engineering tests will continue in the future, implementing an expanded, time-critical AEGIS air engagement capability on additional commercial computing products, building on the experience, tools, and lessons learned to date. Other commercial computer and information transfer technologies will be considered. HiPer-D will evaluate the Navy's new TAC-4 computer in 1996, using a version of the OSF1-MK7 real-time operating system. Network technologies to be examined include high bandwidth switch technologies such as Asynchronous Transfer Mode, Fibre Channel, and Myrinet. Switch technologies—when coupled with new, low-latency protocols—provide point-to-point connectivity for the duration of message transfers, thus reducing latencies well below what is possible with shared media networks such as Ethernet and FDDI. Figure 13 illustrates a future HiPer-D test configuration incorporating these technologies.

Application-to-application message delivery times of less than 100 ms appear achievable with some switch-based LANs. That is faster by an order of magnitude than what most commercial networks and communication protocols deliver in practice today. This is close to the same order of magnitude as context switch times inside computers—the time it takes the operating system to switch between tasks within the computer. In the future,

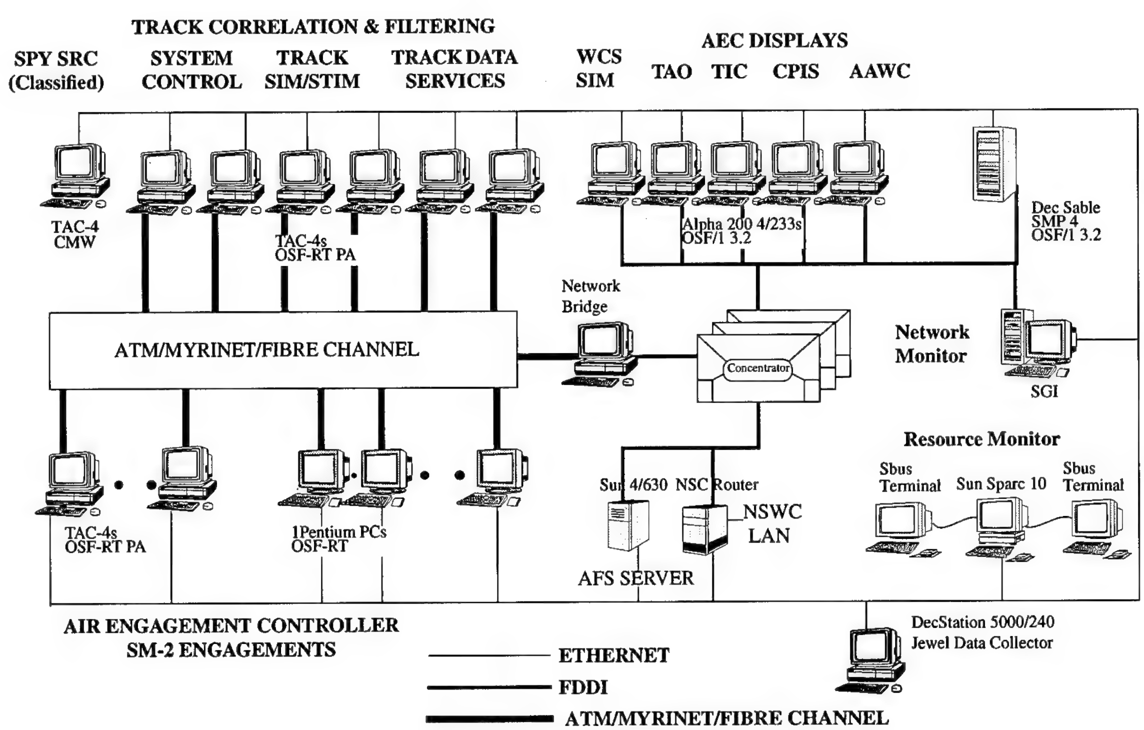


Figure 13. *Future HiPer-D tests will explore the use of the Navy's tactical advanced computers and new commercial switch-based network technologies.*

communication among computers may be only slightly more expensive than communication between tasks within a computer now. This key technological barrier must be overcome to make high-performance, real-time distributed computing a reality. Distributed scheduling is another key to successful distributed computing, because it allows processes in several computers to act in a coordinated fashion while working on an overall computing problem.

The information to be gained in future experiments and engineering tests of the type conducted in March 1994 and May 1995 will continue to define the future for AEGIS and for other future combatants. A vision of that future is illustrated in Figure 14. Eventually, the ship's computing resources may be viewed in the same way that traditional hull, mechanical, and electrical resources are viewed today; i.e., as part of the ship's infrastructure or "hotel services." Computing resources such as these would provide a genuinely new option for ship automation, bringing to reality the idea of "ubiquitous computing," that is, the ability to compute wherever and whenever needed to meet the threat. While this ambitious goal will not be achieved for the initial COTS-based distributed processing implementation now under investigation for AEGIS Baseline 6, Phase 2A, it is a realistic and viable possibility for incorporation into the Surface Combatant - 21st Century (SC-21) now being planned in NSWCDD's Surface Combatant Engineering Center.

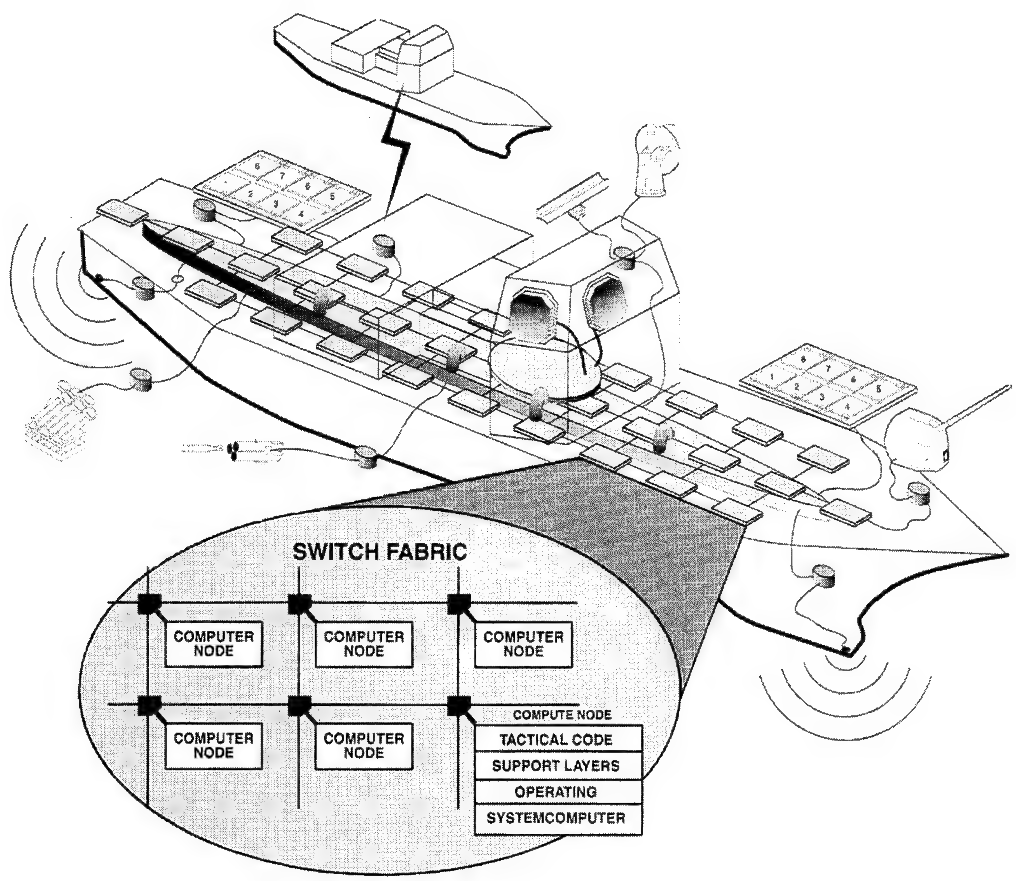
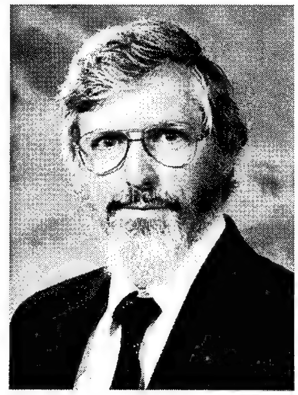


Figure 14. *A vision for future shipboard computing.*

References

1. SecDef ltr (W. J. Perry); Specifications and Standards—A New Way of Doing Business, dated 29 Jun 1994.
2. HiPer-D Computer and Connectivity Architecture Working Group, *Aegis Advanced Baseline Computer System Architecture Alternatives—Version 1.1,* HiPer-D Program, 19 Feb 1993.
3. HiPer-D Demonstration Team, *High-Performance Distributed Computing Integrated Demonstration One Report*, 6 Sep 1994.
4. Geary, J. and Masters, M.W., Investigating New Computing Technologies for Shipboard Combat Systems, Vol. 107, No. 3, May 1995, pp. 253-265.

The Author



MICHAEL W. MASTERS is the System Engineer for NSWCDD's HiPer-D team. He has participated in defining and implementing technology demonstrations based on applying commercial computing technologies to AEGIS-ship Command and Decision applications. He also participated in a top-down system engineering effort that defined and evaluated alternative future combat system architectures based on new computer technologies. Mr. Masters has been employed at NSWCDD since June 1966 as a Mathematician and Computer Engineer. He has worked on many system engineering and software technology projects related to Navy systems. These include: AEGIS; the TOMAHAWK Weapon System; the Combat System Architecture Program; the Integrated Combat System Test Facility; the FBM Program; and the MARK 68 Gun Fire Control System. Mr. Masters graduated from David Lipscomb College, Nashville, Tennessee, in 1966 with a B.A. degree in mathematics with a minor in physics. Since that time, he has taken graduate courses at the University of Tennessee and Virginia Polytechnic Institute.

Molecular Computing: Application of Biomolecules for Information Processing

Ann E. Tate, Jennifer L. Boyd, David W. Cullin, and Robert A. Brizzolara

Due to the computing and signal-processing intensive nature of present and future naval combat systems, the Naval Surface Warfare Center, Dahlgren Division (NSWCDD) has invested in an area of research known as ***molecular computing****. Our team is investigating the development and application of biomaterials with desired properties for use in computational architectures. This article lays a foundation for why biomaterials are chosen and where this research effort fits in the overall scheme of computational efforts here at NSWCDD. Following this, we describe group research efforts in using the tools of biotechnology and genetic engineering to develop these materials. The section after this description is dedicated to characterizing and utilizing the optical properties of these materials. The article concludes with a description of group research efforts in the area of nanofabrication, a relatively new technology that will become vital if we are to realize true molecular-scale computing and electronic devices.*

Introduction

NSWCDD has a strong history and involvement in science and technology areas associated with the advancement of information sciences for Navy systems. As is evident from the articles in this issue, investigations range from theoretical research in quantum computing to applied engineering of the complex cruise missile weapon control systems. This article summarizes ongoing basic and applied research directed toward advances in computing and information processing for significant improvements in the control elements of combat and weapon systems.

Most elements of a Navy combat system are computing intensive. The combat system itself is a highly interconnected network of information processing subsystems that have evolved over decades of engineering development. As today's computer technology pushes us toward a hybrid system integrating parallel and serial processing architectures, it is important to assess where a major advance in computing could potentially have the most positive impact on performance of the overall combat system. It is our conjecture that performance of control systems processing could be significantly improved by shifting most of the data processing to the periphery of the combat system, i.e., out to the individual sensors and weapons. Figure 1 illustrates this scheme where command and control systems would receive preprocessed information for executing critical decision and scheduling functions and would not expend central processing unit (CPU) cycles crunching raw data.

Indeed, the human brain and our sensory systems work on this principle of distributed processing. If we examine the distribution of image processing tasks in vision, the eye, acting as a sensor, transmits a preformed edge-enhanced image via the optic nerve to the visual cortex of the brain. The brain then must match or correlate the input image to a stored image in order for us to recognize the object. A

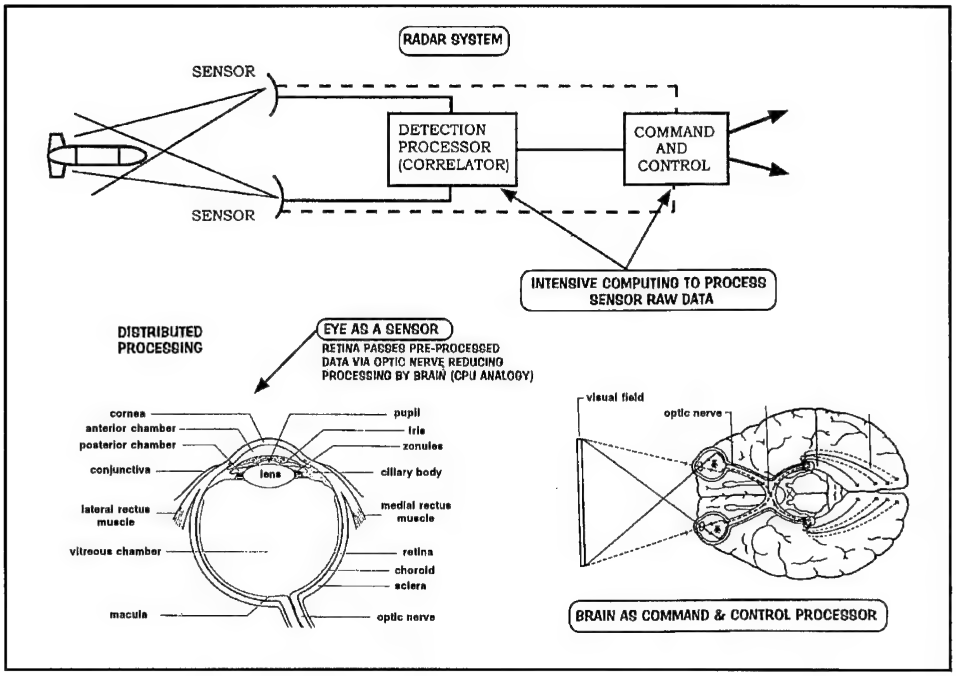


Figure 1. *Analogy of Human Sensory System to a Radar Detection System.*

significant amount of processing occurs in the retina off-loading processing performed by the CPU-like brain.

Research conducted in the molecular computing project looks at coupling the inherently fast response times of selected biomaterials with the parallel addressing power of light to generate smaller, cheaper, and faster information storage devices. The longer term goal is to construct true nanoscale molecular computing elements. This project represents a high-risk investment with enormous payoff potential.

Why Biomaterials?

We chose biomaterials because biomolecules perform powerful information processing functions within cells. They store, copy, and translate coded information into useful products that sustain life. There are two classes of biomolecules prominently involved in information processing found in all living organisms. These are proteins and the nucleic acids, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). Proteins are responsible for many of the functions performed in the cells of living organisms, such as receiving and processing both internal and external information. Proteins also serve as structural elements to maintain form within organisms. DNA and RNA together provide the complete instruction set for all of life. On average, a replica of the complete instruction set is made in one hour during the S phase of the cell cycle. For humans, this equates to copying the information stored in our twenty-three chromosomes, which contain approximately 3×10^9 base pairs of information. It is this coded information in the DNA base pairs that is then translated to make individual subunits called amino acids, which are assembled to generate a protein. It takes a triplet of these base pairs to form one codon or *word* that defines the specific code for a specific amino acid subunit of a protein. It takes about 1200 DNA bases to define the code for a single protein. Consequently, all of the information needed to generate a human being is stored within the nucleus of a single cell less than 50 μm in diameter.

It is this ability to store and process information on a very small scale that was part of the motivation for looking at biomolecules as both a model and a materials source for advances in computing and information processing. These biomolecules have been evolving for billions of years to perform processing within cells in a highly specialized and optimal manner. We want to use some of these properties that individual molecules possess to perform processing in an environment different from a cell and to perform processing tasks useful for our applications. This

can be thought of as an artificial natural selection. Through biotechnology, we have the tools to alter both the environment and the chemical makeup of these molecules and optimize them for our specific applications.

Just as digital computers store information and then use instruction sets to manipulate data and process information, cells use DNA to store information and proteins to translate the information into useful products, namely other proteins, that perform a variety of functions important to intercellular and intracellular processing. These functions include:

- Signal transport
- Signal generation
- Signal transduction
- Energy transformation
- Feedback loops
- Cell repair
- Immunological response

We can consider some of the characteristics of biomolecules that might be useful for extension into new molecular computing constructs. Typically, biomolecules are relatively small compared to semiconductor components. A biomolecule capable of performing some function is usually several nanometers in size as compared to components that are one micrometer on a semiconductor chip. This aspect could potentially reduce the size of components within processing devices. The speed with which these molecules react or mediate reactions varies from femtoseconds (10^{-15} s) to days. Naturally, the ultrafast reactions (10^{-12} s to 10^{-15} s) would be useful in fabricating high-speed switches.

Biomolecules are typically very complex, which provides a richness of structure that allows these molecules to perform highly specialized functions. For example, hemoglobin is a very complex macromolecule that transports oxygen in our bloodstream. Its structure is three-dimensional and actually changes slightly as the hemoglobin functions by transporting oxygen. When the hemoglobin picks up oxygen in the lungs, it assumes one shape or conformation. When it releases oxygen to tissues, it assumes a second, slightly different, shape. This is called a conformational change within the protein. The protein is using a change in its overall shape to perform a function. This shape theme is also very important for the recognition of a specific molecule when a protein catalyzes a reaction that forms a useful product within the cell, such as glucose, the primary fuel source in cells. Several control mechanisms for activating or deactivating reactions within cells use changes to the overall shape of the molecules to inhibit a reaction or to drive a reaction forward.

Within this complex structure, several repeating motifs are found in biomolecules. Two of the more prominent are the α-helix and β-sheets, which are created through hydrogen bonding and van der Waals interactions. The first motif is the familiar spiraling coil typical of DNA and the second motif, having the appearance of alternating pleats, is typically found in proteins. These recurrent structural motifs point to a consistent mode of information processing and signaling within cells that could best be described as ***shape-based*** processing.

Now, let's contrast "computing" characteristics between digital processing and biomolecular processing. We can list four fundamental characteristics of processing. First, digital machines are structurally programmable. The structure of the machine (state of gates, switches) is independent of the function being processed. The software or program can be changed to execute a different program that performs a different function. Digital machines are highly programmable. This costs in terms of inefficiency, but pays in terms of programmability. In a protein, the function being performed is inseparable from the structure or state of the molecule. This can be considered shape-based processing—if the structure of the protein is changed, the function is also changed. As with the previous example of hemoglobin, one shape functions to take up oxygen, and a second shape releases oxygen. Hemoglobin can alternate between these two states repeatedly. Such shape-based conformational changes are critical to regulating the processing activity of most proteins.

By using the biotechnology tool of genetic engineering, specific alterations can be made to change the shape and chemistry of proteins. In this manner, we can redesign some of their characteristics in an effort to optimize desirable properties for information storage and processing. If the shape of the protein is changed too much, it will denature and lose all functionality. However,

small changes can be made to the protein to alter its function without losing activity or function.

Continuing with our contrast of processing between digital machines and biomolecules, we realize that digital machines compute symbolically, while biomolecules process information in a dynamic and physical mode. Biomolecules change their state or shape based on their environment and thereby change their function. If we look at what makes a digital machine "smart," it is dependent on the intelligence of the programmer for optimization. Biomolecules depend on the process of evolution for optimization. Over billions of years, Mother Nature perpetuates the slight changes that produce some advantage. The speed at which visual receptors respond to light was recently measured at Berkeley,[1] and it is one of the fastest photochemical reactions ever measured. It is a scientific challenge in itself to be able to resolve events with femtosecond (10^{-15}) resolution. We find that the basic photochemistry in the receptor molecule is similar for all organisms that use vision for survival. Clearly, nature has optimized and perpetuated a supremely efficient photoprocessing system.

Lastly, we know that digital machines are generalized. We use our personal computers (PCs) to execute many different programs that perform various processing functions. In contrast, biomolecules are highly specialized. They have been optimized over billions of years of evolution to perform a single function very efficiently.[2] If we revisit the example of our eye and vision, the receptor molecule in the retina is a biomolecule called rhodopsin. This biomolecule is extremely efficient at capturing light. The initial response occurs in 200 fs. The receptors transduce the photosignal into an electrical signal used by two other cell types in the retina that perform spatio-temporal processing and send an edge-enhanced image to the brain. These receptors are highly specialized for receiving a photosignal only. They would not be able to receive or process acoustic or tactile signals. Although specialized, their ability to detect roughly 70 percent of all visible photons in 200 fs is very optimized. In contrast to generalized processing in digital machines, molecular processing is highly specialized.

Next, we describe the results and progress in several scientific areas. We have just described the motivation for selecting biomolecules as a materials source for computing applications and briefly contrasted digital and biomolecular processing. We will now discuss genetic engineering techniques we are employing to modify biomaterials for computing applications. Following that section, we will describe how these altered materials are characterized and the introduction of a novel computer memory architecture using biomaterials and light to store terabytes in a cubic centimeter. The capability to manipulate these materials for the fabrication of nanoscale components must be developed. The last section of this article discusses progress in the area of nanofabrication. The development of this device fabrication technology is critical to future molecular electronic device applications.

Biomaterial Design and Modification

The concept of molecular computing is based on creating molecular scale electronic devices to be used in computing applications. There are several sources of materials that are the subject of research in this field. Silicon-based semiconductor materials could and are being further miniaturized and are rapidly approaching molecular size. Another approach would be to use these silicon or other semiconductor-based materials in fashioning biomimetic devices. For example, an artificial neural network built of silicon but architecturally similar to neural cell networks could be created. Potentially, the most promising source of materials is nature itself—these materials are the focus of our research efforts. To use the biomaterials that nature has optimized over millions of years and designed for very specific functions taps into an entire world, barely explored. Cellular communication on the molecular level is a vastly complex and intricate network performed solely by proteins. Proteins are combinations of 20 different amino acid building blocks that fold into specific shapes defined by their chemical interactions. Those specific shapes define the protein's function. Each protein has evolved to possess a shape that allows it to perform its function in the most efficient way possible.[3] It is this idea of functionality being inherent in structure that has prompted us to investigate proteins. We wish to understand their highly evolved shape-based processing and investigate how to exploit them for computational tasks.

Our approach to molecular computing involves using proteins in conjunction with other materials and devices to perform tasks they were not specifically evolved for. Therefore, the characteristics that we wish this protein to have, for computing applications, may be slightly different than those for which it was originally designed through the evolutionary process. There are two general methods for modifying the protein's characteristics. The first is to put it in the presence of certain chemicals so that its properties are changed based on a chemically dictated environment. This often results in the inability to fine-tune a process. Another alternative is to genetically engineer the protein such that we substitute one of its amino acid building blocks for another and consequently modify the protein. This is done at the level of the DNA that codes for the appropriate amino acids, hence the term *genetic engineering*.

The protein on which we have focused the majority of our attention is bacteriorhodopsin (BR). BR is a membrane-bound protein produced naturally in *Halobacterium halobium*, an archaebacterium found in concentrated sodium chloride environments. The protein, BR, is made up of seven alpha-helical amino acid moieties forming a channel through the membrane of the bacteria. Situated in the protein pocket is an all-transretinal chromophore attached via a Schiff-base to the 216th amino acid, lysine. Upon absorption of optical frequency light centered at 570 nm, the protein undergoes a complex photocycle culminating in the movement of a proton from the intracellular side of the protein to the extracellular side. Various states that are generated during this photocycle are what we attempt to make use of in optical processing architectures employing BR. The photocycle is shown schematically in Figure 2 and will be described briefly. The initial state (B in Figure 2) absorbs a photon of light, and a transition occurs to an electronically excited state of B. Each state, represented by letters in Figure 2, is a distinct conformationally different species of the initial B state. Following the absorption of light, the electronically excited B state decays and, in approximately 70 percent of all cases, the photocycle progresses through the J intermediate (thought to be a vibrationally excited K state

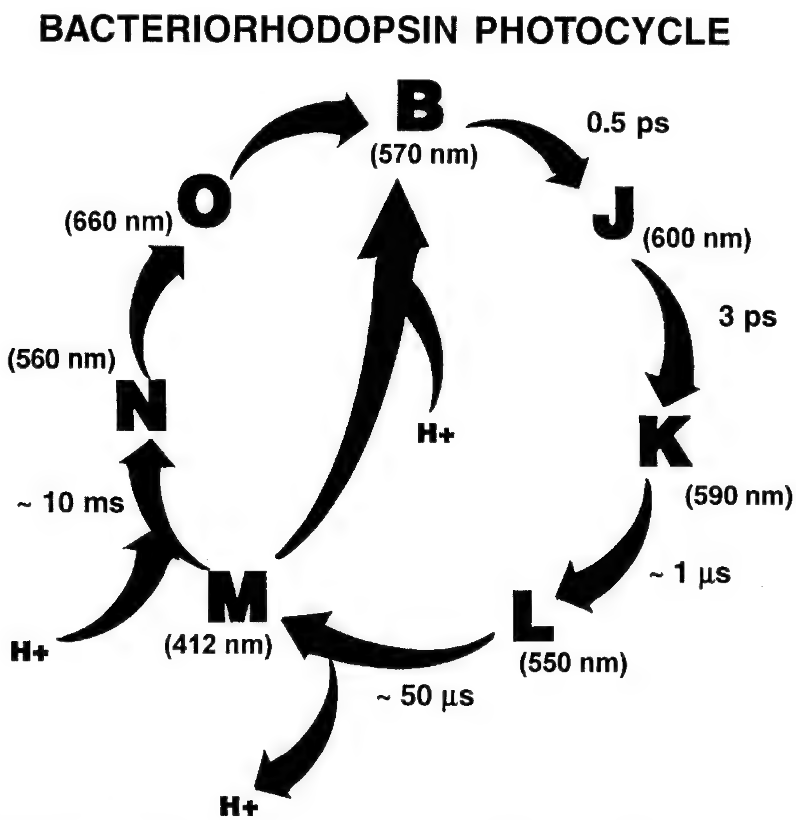


Figure 2. *Diagram of the BR photocycle. This diagram represents the energy transfer (state changes) through the protein that occurs after the chromophore absorbs a photon of light.*

species), and on to the K intermediate. The K state is identical to the B state except there is a rotation about a carbon-carbon bond changing the chromophore to a 13-cis configuration. There is a great deal of energy (approximately 12 Kcal) left over after the molecule undergoes the rotation, and this drives more relaxation in the protein, which is manifested in slight motion of the protein backbone surrounding the chromophore (L state).

About 50 μs after photocycle initiation, the M state is formed, which is the only state in the photocycle where the Schiff-base linkage is deprotonated. Because this state is so chemically distinct from the rest of the photocycle, it is this state, with the exception of the initial B state, that has been studied most extensively. It also represents the second state in any holographic or two-state switch applications that have been suggested. Its lifetime has also been shown to be dramatically affected through selected amino acid substitutions, and it has been shown to be susceptible to chemical alteration. After the proton is pumped, the Schiff-base is reprotonated (N state), and the molecule thermally relaxes back to the initial state (B state). It is also worth mentioning that it is possible to drive any of the described states back to the ground state without allowing thermal relaxation to occur. This can be done through the absorption of a photon of the correct frequency. Each state has its own distinct absorption profile.

As described, the chromophore absorbs photons, converting it from all-trans to 13-cis, and the transfer of that energy to the surrounding helices of the protein causes them to undergo a series of conformational changes. These conformational changes drive the transport of hydrogen ions from the inside of the bacterial cell to the outside, generating a proton gradient. This gradient, in turn, drives energy synthesis for the cell under conditions of low oxygen concentration.[4] For information processing applications, we are interested in researching both the photovoltaic and photochromic properties of BR and how subtle changes in the amino acid makeup of the protein can alter those properties.

Much work has been done towards genetic modification of BR including amino acid substitution studies and the development of expression systems in *E. coli* and *H. halobium.*[4-6] For example, the most prevalent mutant in the literature is D96N. This BR variant was derived by replacing the 96th amino acid, aspartic acid, with asparagine. The most noticeable effect of this change is an increase in the M state lifetime of the protein. The change reflects a decreased capacity for proton transfer through the protein channel and, therefore, BR remains in the M state of the photocycle for an increased length of time. This alteration has allowed progress in researching the holographic recording abilities of BR.[5] Another example of amino acid substitution with BR is S35C, where the 35th amino acid is changed from a serine to a cysteine. The applications of this modification are discussed below. Our genetic modification work focuses on expanding these systems and exploring new mutants and their application to information processing. The genetic studies involve making amino acid changes through site-directed mutagenesis of the DNA encoding the protein, expression of that DNA to protein, and the isolation of protein from the bacteria cell. Once isolated, the altered protein can then be spectroscopically studied to determine the effect of the change to the protein and whether or not that change is useful for device applications.

In addition to our study of this protein, we are also exploring other biomaterials for their utility in device construction. For example, the use of antibodies (another protein) is being explored as a mechanism for attaching and patterning BR to solid surfaces for device fabrication (Figure 3). Monoclonal antibodies are being designed and developed to orient and attach BR to surfaces such as polystyrene and glass. An alternative method to antibody attachment is site-directed mutagenesis. In this example, the S35C mutant described previously is used. The cysteine contains a sulfhydryl group that will directly bond to a gold-plated surface. There are structural and patterning advantages and disadvantages to both methods of patterning. For example, the antibody may make BR patterning easier through ultraviolet (UV) photochemistry. The direct contact of the surface with the BR may increase chances for the retention of protein function. These methods will be discussed further below.

In summary, the power of genetic engineering and biotechnology is being used to slightly alter the functionality of biomolecules and optimize them for use in information processing applications.

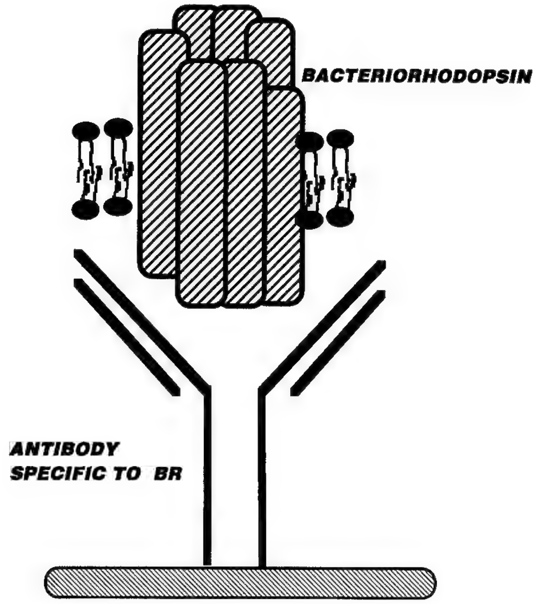


Figure 3. *In the patterning scheme of immunological anchoring, BR is attached to a solid surface through interaction with an antibody molecule.*

Optical Characterization and Application of Biomaterials

When one contemplates the characteristics of various materials that make them attractive candidates for use in optical computing applications, there are several figures of merit that need to be considered. Among these are their sensitivity to light of a given wavelength; their stability under illumination, or cyclicity, before photodegradation; their structural stability; the ability to control state lifetimes as if the material were a two-state switch; and the ability to toggle back and forth between these states quickly.

These characteristics are present or nearly present in the BR protein and, through the use of genetic modification of the protein, we are able to "tweak" those characteristics to our desires. First, the amount of light necessary to activate the BR photocycle is small, as it has been shown that approximately 70 percent of all photons of the proper wavelength that are absorbed will initiate the photocycle.[7] Second, the cyclicity of the protein has been measured in terms of holographic read-write-erase cycles to be greater than 10 million.[8] Third, the protein is stable to about 70°C before it will denature.[7] Fourth, although we have not yet been able to control the M state lifetime completely, we are able to control the thermal decay of the M state of the photocycle through the use of chemical modification as well as genetic manipulation. Lastly, the switch time between the initial state of the photocycle and the M state is on the order of microseconds (quite fast in parallel computing applications), while the transition between the initial state and the K state is a picosecond process. Therefore, naturally occurring BR has the potential to be an ideal optical storage or processing element. We have undertaken the task of making selected modifications to the protein that will allow it to have the "perfect" characteristics for desired applications.

There are several steps in developing modified BR and ultimately utilizing the protein in optical processing applications. We can break this process into several parts:

- The creation of the genetically or chemically modified material
- The characterization of the variants for their optical properties
- The iteration of that process to optimize desired properties
- Their use in novel devices

This section will describe ongoing optical characterization experiments for protein holographic and photochromic properties and take a glimpse at the application of BR in a three-dimensional holographic memory.

One of the many applications suggested for BR is its use as a holographic storage element. Recently, we have been investigating the holographic properties of both genetically and chemically modified thin films of the protein. We have completed a series of experiments that have examined the effect of protein solubilization on the holographic and optical properties of BR.[9] BR is one of the few proteins that can be isolated in a two-dimensional lattice of a protein-lipid complex (often referred to as purple membrane (PM)). Its structure can therefore be studied using powerful diffraction methods.[10] There have been relatively few studies that explore the actual effect of this two-dimensional structure on the photophysics of the protein. This series of experiments attempted to explore the effect of the breakdown of the two-dimensional structure and its effect on the absorptive and holographic properties of PM thin films.

This investigation was carried out by fabricating a series of thin films of BR with varying

amounts of the detergent Triton X-100 contained in them. Previous studies have shown that adding this detergent causes the breakdown of the protein-lipid complex ultimately leading to monomer BR formation. We carried out a series of absorption and holographic spectroscopy experiments that investigated the M state lifetime as a function of detergent added, as well as a series of holographic growth experiments that provided us with information regarding the effect of PM breakdown on holographic sensitivity. The results of the latter series of experiments is shown in Figure 4. This figure plots the holographic sensitivity as a function of Triton X-100 added. As shown, the detergent addition had the effect of increasing the holographic sensitivity to a plateau value around 15 percent Triton/BR ratio. We also found that the holographic (and absorptive) lifetime of the M state after adding Triton X-100 was approximately a factor of two to three longer.

We have also been conducting a series of investigations with the goal of developing a BR species with an extended K state lifetime. This is being carried out by using transient absorption decay and transient Raman spectroscopies with picosecond time resolution. The K state is formed in approximately 3 ps, and its thermal lifetime is on the order of several microseconds. Our aim is to develop a two-state switch with a switching time that would be several orders of magnitude faster than that of the initial (B state) to M state transition.

The initial to K state transition involves the absorption, by the initial state, of a quantum of light, causing the excitation of the initial state to an excited electronic state. The protein then relaxes, and in approximately 70 percent of all cases, a rotation about a bond occurs to form the K state. In this process of absorption, the excess energy remaining after the torsional motion is localized in the chromophore of the protein. It is this excess energy that drives the rest of the motions of the protein and the proton translocation that occurs in the photocycle. We would like to be able to control where that energy goes, such that it will not drive the rest of the photocycle and will be truncated at the K state.

Our strategy is to first understand and ultimately control the excess energy transduction pathways leading from the chromophore to the rest of the protein backbone. We have begun picosecond absorption spectroscopy studies, which allow us to follow the formation and decay of the K state in the wild-type and several different mutant BR species where we have selected specific amino acids that we believe are involved in this energy transfer process. We are also using transient Raman spectroscopy of these variants, which will lead to knowledge about specific molecular vibrations in the protein as the

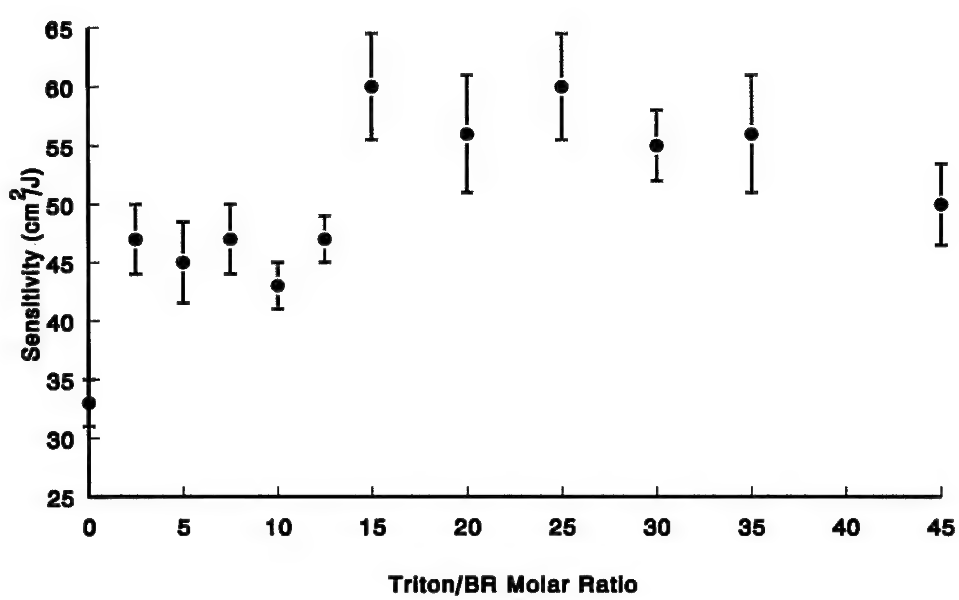


Figure 4. *Plot of measured holographic sensitivity versus Triton X-100/BR molar ratio. It is believed that complete solubilization occurs at a ratio of about 20:1.*

photocycle proceeds. Our intent is to design a new variant BR material with an infinite K state lifetime where one could employ two distinct lasers of different wavelengths to address the initial and K states. This would allow researchers to replace BR employing the initial to M state transition with this much faster material in their optical processing or optical memory applications.

We have also become involved in the development of a three-dimensional holographic memory with thin films of modified BR as the volume storage element, in cooperation with Biological Components Corporation (BCC). It is expected that we will be able to store terabytes in a 3-cm by 500-μm thin film and realize write times of 200 ms/page (130 Mbytes/s) and similar read times of >100 ms/page (or 260 Mbytes/s). Although those types of access times do not seem particularly fast, remember that we are talking about accessing (either writing or reading) entire images in parallel in that time frame.

A schematic representation of the holographic memory architecture is shown in Figure 5 and can be described as follows. This memory is not three-dimensional in the conventional sense in that it is not a cube but rather a thin film. The three dimensions in this case are the normal X and Y directions (spatial multiplexing of the holograms), as well as a third dimension—the angular rotation of the film. This storing of holograms in a rotational dimension is called angular multiplexing and is available to us because the storage and readout of the holographic interference pattern is highly dependent on the angle at which it is stored and the relationship between the write and read angles. Therefore, many holograms are stored on top of each other at a given position (X,Y) on the thin film. The film is rotated to different addressable rotational positions.

Holograms are generated by storing the interference pattern resultant between a plane wave (or reference beam) and the light diffracted or reflected from some object (object beam). In this case, the image (or page of images) is first sent to a spatial light modulator (a liquid crystal projection TV element), which modulates or puts the image onto our object laser beam. These two beams are then overlapped at the BR film surface, and an interference pattern is generated in the overlapping region. In areas of constructive

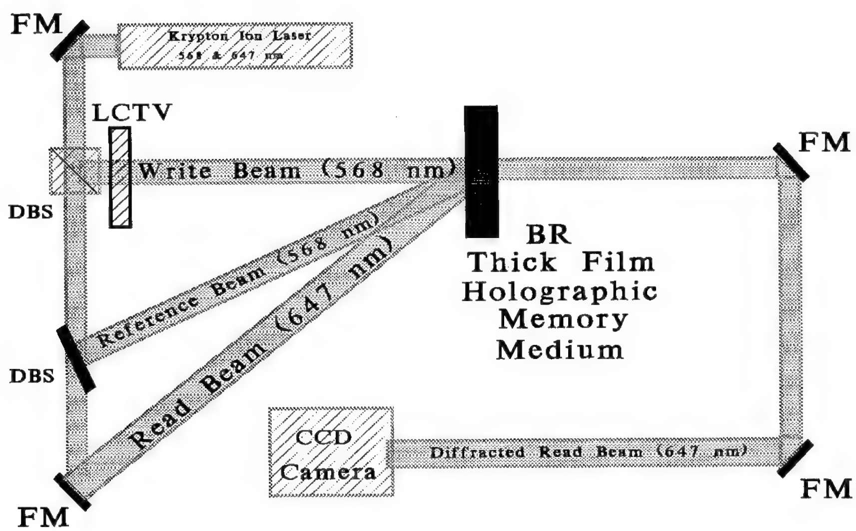


Figure 5. *Schematic diagram of the proposed three-dimensional memory based on a thick-film bacteriorhodopsin holographic medium. Images are introduced through the use of an electronically addressed spatial light modulator (LCTV). The write, read, and reference beams are provided by the dual output of a Krypton ion laser. The 568-nm output serves as the write beam and the 647-nm output serves as the readout beam. The write beam is overlapped with the reference beam creating an interference pattern in the area of the overlap. Interference is stored in the BR film. Stored images can then be read out using the 647-nm read beam aligned at the proper Bragg angle. The readout is then captured on a charge couple device (CCD) array and loaded to a PC or other digital device. FM stands for fixed mirror, and DBS for Dichroic beamsplitter.*

interference, there is light that strikes the BR film and is absorbed, and that region is then converted to M state. In regions of destructive interference, no photochemistry occurs. In this way, we can store the interference pattern, leaving behind alternating regions of 570 nm absorbing initial state and regions of 410 nm absorbing M state. This storage is accomplished in a completely parallel fashion in microseconds. To read out the image, one must irradiate the desired address with the laser wavelength chosen, and the hologram is reconstructed. The speed of the readout process will be limited to the speed of whatever charge couple device (CCD) camera is used to recapture the image.

There are still several research and development questions that need to be answered to make this device successful. The most pressing problem is to make the M state lifetime infinite so that data would not be destroyed through thermal decay processes. The researchers intend to use a combined chemical modification and genetic engineering approach to solve this dilemma. There are also several architectural questions that need to be resolved, as well as the method of interfacing this memory to the digital computer world. If successful, one could see this memory becoming an add-on peripheral to today's computers that could be used as an extremely fast, high-density graphical storage element, not unlike a compact disk read-only memory (CD ROM) except one would easily be able to both read and write to these storage films.

Patterning and Nanofabrication

The extremely fast, efficient electrical response of the BR molecule to light makes it attractive for use as a light-electrical transducer in device applications. Such applications may well occur in novel computational architectures, such as the artificial retina. The utility of BR in such applications depends on the ability to pattern it on a chip in thin films of controlled spatial extent and thickness. The analogy in the silicon world is the lithographic definition of structures and devices on a silicon wafer. Patterning requires the tethering of BR to the substrate at defined attachment points with a controlled molecular orientation, without destroying the light absorbing and proton pumping abilities of the protein.

The artificial retina is an excellent example of the benefits in computational speed and efficiency that can be realized by mimicking computational architectures found in nature. The human retina, located at the back of the eye, functions not only as an image detector, but as an image processor. It performs the processing in a manner different from ordinary computers due to its parallel architecture. Because of this architecture, it is capable of processing visual information in real time. Scientists and engineers are seeking to duplicate the retina's architecture on a chip that might be used for machine or robot vision applications. So far, the bulk of this work has been done on silicon. In the late 1980s, Carver Mead (California Institute of Technology, Pasadena, CA) succeeded in implementing retinal processing functions on a silicon chip.[11] More recently, the Japanese have made great progress in this area.[12]

There would be advantages in implementing artificial retinas using biological materials such as BR rather than silicon. First, the efficiency, speed, and sensitivity of the BR could be exploited. Second, it has been shown by Miyasaka (Fuji, Kanagawa, Japan) and A. Lewis (Hebrew University, Jerusalem) that retinal image processing functions can be implemented using BR, without the associated circuitry that is required in the silicon implementation.[13,14] This gives a power savings as well as conserving real estate on the chip surface.

Another application of the light-electrical transduction of the BR molecule is in the field of nanotechnology. Here, scientists are striving to make devices smaller so that higher integration densities on a chip can be achieved. Advances in semiconductor lithography are making this possible. An alternative approach is to use nanometer scale devices that already exist in nature; i.e., functional biological molecules such as BR. A single BR molecule is roughly 4 x 5 x 6 nm in dimension compared with the smallest gate size achieved on semiconductors to date, which is on the order of hundreds of nanometers. To nanofabricate a

discrete light detector, it may be advantageous to use BR rather than silicon. This would convert the problem from one of fabricating a nanometer scale device to one of tethering a "prefabricated" device (the BR molecule) to the chip surface. It may seem unrealistic to think of manipulating individual molecules on a surface; however, in fact, this technology already exists. In the late 1980s, D. Eigler at IBM demonstrated the ability to manipulate individual atoms on a surface using a device called a Scanning Tunneling Microscope.[15] The challenge today is to understand and control the process of atomic and molecular manipulation so that this may be done quickly and reliably with a variety of materials.

In the Molecular Computing Group at NSWCDD, methods for patterning BR on a surface are being developed. Several approaches are being pursued—all are designed to be compatible with nanometer-scale patterning. In the first approach, the substrate surface is chemically modified in order to enhance the adsorption of BR with a preferred orientation. More precisely, the BR is adsorbed on a surface in its native environment, the PM. Since all BR molecules in one fragment of PM have the same orientation, if the PM patches can be deposited with the same side facing the substrate surface, all BR molecules in that film will have the same orientation. Ultimately, the PM fragments can be broken up into smaller pieces to facilitate the fabrication of smaller structures. In a recent publication, we have shown that surfaces terminated by different chemical functionalities have differing affinities for the PM.[16] Presently, we are exploring ways of controlling the orientation of the PM by exploiting chemical and electrostatic interactions between the PM and the surface.

Another patterning method, which utilizes immunological techniques, is being explored in a collaboration between the NSWCDD Molecular Computing Group and Professor G. Vasta at the Center of Marine Biotechnology at the University of Maryland. In this approach, an antibody to the BR molecule is isolated. A thin film of the antibody molecules is created on a substrate surface using well-known techniques from molecular biology. These antibody molecules bind to the BR at a specific site on the BR molecule; thus, the BR is bound to the substrate surface in a specific orientation. Figure 6 shows an Atomic Force Microscope image of a thin film of anti-BR that has been exposed to PM fragments. The PM fragments can be clearly seen. Figure 7 shows a thin film of antibody molecules that are not specific to BR and have been exposed to PM fragments. No PM fragments have been bound

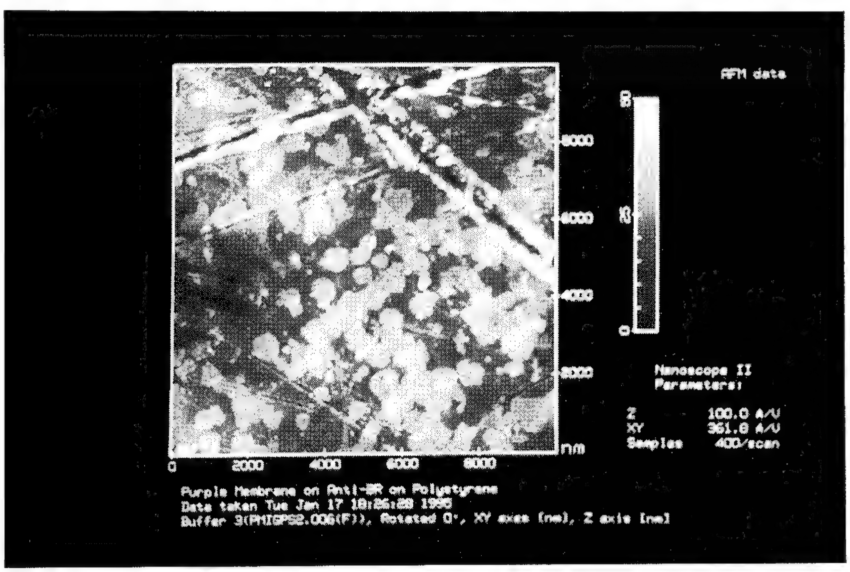


Figure 6. *10000 nm x 10000 nm AFM image of an antibody thin film specific to bacteriorhodopsin on a polystyrene substrate. This surface was exposed to purple membrane. The image shows that membrane fragments have been bound to the surface. The membrane fragments are roughly circular in shape, 500 to 1000 nm in diameter, and 5 nm in height. The straight lines in the image are features in the polystyrene substrate.*

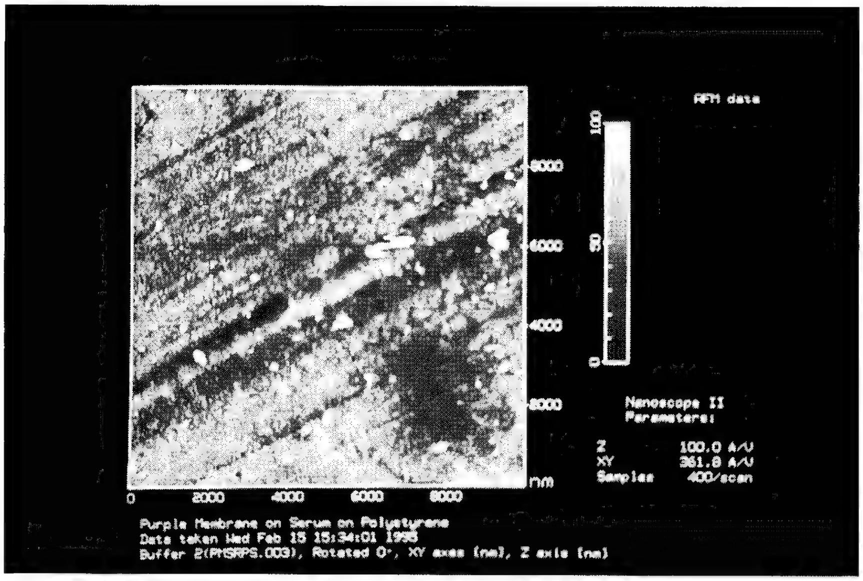


Figure 7. *10000 nm x 10000 nm AFM image of an antibody thin film not specific to bacteriorhodopsin adsorbed on a polystyrene surface. The surface was exposed to purple membrane. No purple membrane fragments have been bound by this surface.*

by this surface. Thus, the anti-BR surface in Figure 6 has a specific affinity for BR. Exposure to UV light induces chemical changes in the antibody molecules that destroy this specificity: this may be useful for patterning the PM on a chip surface. It is fascinating to think that principles from the human immune system might one day be used to fabricate computer chips!

A third patterning method utilizes methods from genetic engineering to modify the BR molecule to facilitate its attachment to a surface. Note that in the first two techniques, the surface was modified to promote attachment of the BR. In this technique, the BR molecule itself is modified using genetic engineering techniques. Specifically, one amino acid residue in the BR protein is replaced by a cysteine residue. Cysteine is an amino acid that possesses a sulfhydryl group, which has a very strong affinity for certain metals such as gold, silver, and copper. The genetically modified BR molecule will bind chemically to these metal surfaces. This can be verified through the use of x-ray photoelectron spectroscopy (XPS), a surface analysis technique that reveals the chemical composition of a sample surface. Figure 8 shows the XPS sulfur spectrum of wild-type (not genetically modified) PM. This is the "normal" spectrum of PM and shows the expected sulfur chemistries. The spectrum in Figure 9 shows the sulfur spectrum from fragments of PM containing genetically modified BR that have been deposited on a gold surface. In addition to the peaks in Figure 8, there is an additional peak at 162 eV that is indicative of sulfur bound to gold. Thus, the BR molecule has been genetically modified to allow it to chemically bond to the gold surface.

By using techniques from scientific fields as diverse as surface chemistry, genetic engineering, and immunology, we are learning how to attach BR to a surface with precise control over the position and orientation of the molecules. This technology will not only extend the present trend toward higher integration densities on chips, it will facilitate a new generation of computers, such as the artificial retina, that will derive their tremendous computational power not only from increased chip integration densities, but also from their unique architectures.

Conclusions

In this article, we have described ongoing basic and applied research being conducted at NSWCDD directed toward improving the speed and efficiency of computing and information processing for Navy combat systems. We have

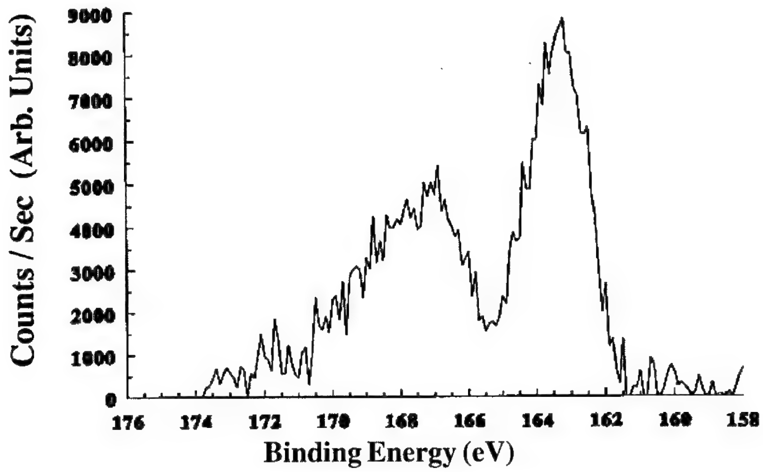


Figure 8. *XPS spectrum showing sulfur peaks due to wild-type purple membrane. The peak at 163 eV is due to the thioether linkage in the methionine residue in the bacteriorhodopsin protein. The peak at 168 eV is due to sulfate-containing lipids in the purple membrane.*

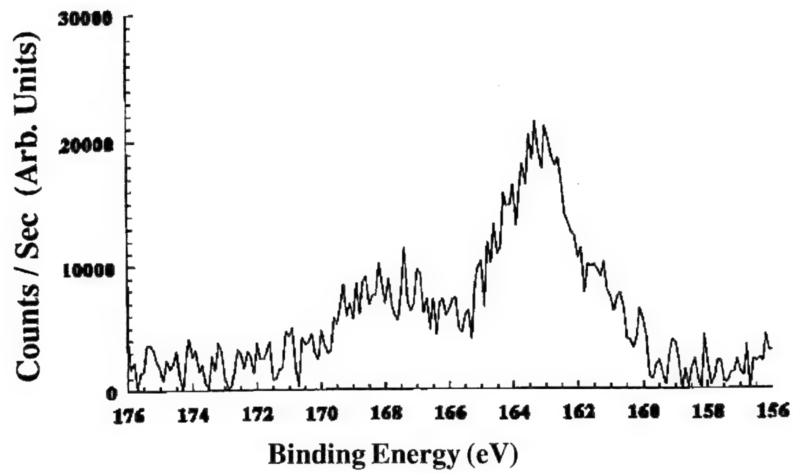


Figure 9. *XPS spectrum showing sulfur peaks due to the genetically modified purple membrane bound to a gold substrate. This spectrum displays both peaks present in Figure 6 plus a peak at 161 eV which is due to gold, indicating the purple membrane has been chemically bonded to the surface.*

examined aspects of biological computational systems as a means of realizing these performance gains, specifically biomimetic architectures and biological materials. The inherently parallel computational architectures found in biological systems are well-suited for many computations performed by Navy systems. Biological materials, such as proteins, have characteristics, optimized by nature, that make them attractive for application in computational devices. Furthermore, these characteristics can be optimized through genetic engineering. A good example is the extremely fast, efficient light-electrical transduction of BR.

Before these concepts can become reality, specific technical issues must be resolved. This article has described some of those issues and the work performed at NSWCDD directed toward resolving those issues. For example, increasing the lifetime of the BR M-state is a critical element in utilizing BR in a memory application. Second, patterning BR on a surface with a controlled orientation must be accomplished to employ it in nanometer-scale applications, or applications in which larger structures are required, such as artificial retinas. Genetic engineering techniques will likely play a crucial role in resolving both issues.

The most significant improvements in the information processing capabilities of Navy systems during the next decades likely will be achieved by: (1) shifting much of the data

processing to the periphery of the combat system, i.e., the sensors and weapons, and (2) utilizing alternative computational architectures. We expect that both of these will be accomplished using elements of biological computational systems.

Acknowledgments

The authors wish to thank Professor Robert Birge, Professor Gerardo Vasta, and Professor Richard Needleman for their insightful discussion, collaborations, and assistance in carrying out this research. This work was supported by grant number N00014-94-WX23079 from the Office of Naval Research, the NSWCDD Independent Research Program, and the NSWCDD Science and Technology Office.

References

1. Wang, Q. and Shank, C.V., "Vibrationally Coherent Photochemistry in the Femtosecond Primary Event of Vision," *Science*, Vol. 266, 21 Oct 1994, pp. 422-424.
2. Conrad, M., *Advances in Computers*, Academic Press, Inc., Vol. 31, 1990, pp. 246-256.
3. Darnell, J.; Lodish, H.; and Baltimore, D., "Molecules in Cells," in *Molecular Cell Biology*, Scientific American Books, 1990, New York, pp. 43-84.
4. Khorana, H. G., "Bacteriorhodopsin, a Membrane Protein That Uses Light to Translocate Protons," *The Journal of Biological Chemistry*, Vol. 263, 5 Jun 1988, pp. 7439-7442.
5. Hampp, N.; Brauchle, C.; and Oesterhelt, D., "Mutated Bacteriorhodopsins: Competitive Materials for Optical Information Processing?" *MRS Bulletin*, Vol. XVII, No. 11, 1992, pp. 56-60.
6. Ni, B., et al., "An efficient system for the synthesis of bacteriorhodopsin in halobacterium halobium," *Gene*, Vol. 90, 1990, pp. 169-172.

Glossary

α-helix: A three-dimensional spiral motif found in DNA and proteins formed from hydrogen bond and van der Waals interactions between repeating subunits

β-sheet: A three-dimensional pleated motif found in proteins stabilized by hydrogen bonding between subunits

antibody: A protein produced in response to an invading agent or antigen to protect the organism against the invading agent; the antibody is very specific for a particular antigen

base pair: A complementary matching of nucleotide bases in DNA based on hydrogen bonding that provides a mechanism for nucleic acid bases to recognize one another

biomimetic: An approach where computational architectures mimic biological systems

chromophore: The part of a molecule that absorbs at a specific wavelength of light

chromosome: The structural unit of genetic material in a cell consisting of DNA and histones

cis: A structural isomeric form, part of cis-trans isomerization, meaning *on this side*

codon: A triplet of nucleotides that serves as the coding unit for assembling amino acid subunits into a protein

conformation: The three-dimensional shape of a protein or nucleic acid

cysteine: A nonpolar amino acid containing a sulfur atom in reduced form (-SH)

denature: A loss of the three-dimensional structure of a macromolecule that usually results in loss of biological activity

enzyme: A protein that functions as a biological catalyst acting on specific substrates converted into products

gene: A section of DNA on a chromosome that codes for a specific protein

genetic engineering: A term used to describe an area of biotechnology where techniques are employed to alter the genetic code

holography: A process by which the interference pattern generated by the interaction of a plane light wave with an object light wave is stored in a medium

hydrogen bond: A weak attractive interaction between a hydrogen atom and an uncharged polar electronegative atom

immunology: The study of the immune system, including antigens and antibody responses

lipid: Amphipathic molecules that are a major component of membrane structure

moiety: A functional unit of a biological molecule

monoclonal antibody: Homogeneous immuno-globulin derived from a single clone of cells

mutagenesis: The process of introducing natural or artificial changes to the genetic code

nanoscale: 10^{-9} meters, seconds, etc.

nucleic acid: Polymers consisting of a nitrogenous base linked to a sugar and a phosphate group—deoxyribonucleic acid (DNA) and ribonucleic acid (RNA)

photochromic: Having the property of changing color upon absorption of different wavelengths of light

protein: A macromolecule consisting of one or more polypeptides folded into a conformation specified by the sequence of amino acid subunits and functioning as an enzyme, a hormone, an antibody, or a structural component

serine: An uncharged polar amino acid

site-directed mutagenesis: Very specific alteration of a single nucleotide base in DNA

sulfhydryl group: A group containing both a sulfur and a hydrogen atom

trans: A structural isomeric form, part of cis-trans isomerization, meaning *across*

van der Waals: A weak, attractive, or repulsive interaction between atoms based on chance inequalities in the distribution of electrons in a bond

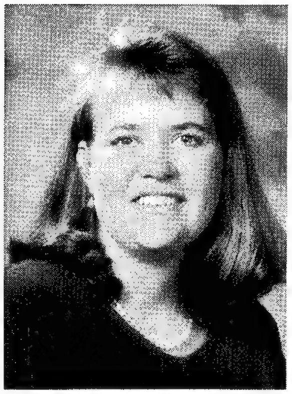
7. Birge, R. R., "Nature of the Primary Photochemical Event in Rhodopsin and Bacteriorhodopsin," *Biochimica et Biophysica Acta*, Vol. 1016, 1990, p. 293.
8. Oesterhelt, D.; Brauchle, C.; and Hampp, H., "Bacteriorhodopsin: A Biological Material for Information Processing," *Quarterly Reviews of Biophysics*, Vol. 24, No. 4, 1991, p. 425.
9. Cullin, D. W.; Vsevolodov, N. N.; Dyukova, T. V.; and Wheetal, H. H., "Optical Properties of Triton X-100 Treated Purple Membranes Embedded in Gelatin Films," *Supramolecular Science*, Vol. 2, 1995, pp. 25-32.
10. Henderson, R.; Baldwin, J. M.; Ceska, T. A.; Zemlin, F.; Beckman, E.; and Downing, K. H., "Model for the Structure of Bacteriorhodopsin Based on High-Resolution Electron Cryo Microscopy," *J. Mol. Biol.*, Vol. 213, 1990, p. 899.
11. Mead, C. and Mahowald, M., "A Silicon Model of Early Visual Processing," *Neural Networks,* Vol. 1, 1988, pp. 91-97.
12. Nitta, Y.; Ohta, J.; Tai, S.; and Kyuma, K., "Monolithic Integration of Optical Neurochip with Variable Sensitivity Photodetector," *IEEE Photonics Technology Letters,* Vol. 5, 1993, pp. 67-69.
13. Takei, H.; Lewis, A.; Chen, Z.; and Nebenzahl, I., "Implementing Receptive Fields with Excitatory and Inhibitory Responses of Bacteriorhodopsin Films," *Applied Optics*, Vol. 30, 1991, pp. 500-509.
14. Koyama, K.; Yamaguchi, N.; and Miyasaka, T., "Antibody-Mediated Bacteriorhodopsin Orientation for Molecular Device Architectures." *Science*, Vol. 265, 1994, pp. 762-765.
15. Eigler, D. and Schweizer, E., "Positioning Single Atoms with a Low Temperature Scanning Tunneling Microscope," *Nature*, Vol. 344, 1990, p. 524.
16. Brizzolara, R. and Beard, B., "Control of Purple Membrane Adsorption to a Glass Surface Using Self-Assembled Monolayers," *Journal of Vacuum Science and Technology,* Vol. A12, 1994, pp. 2981-2987.

The Authors

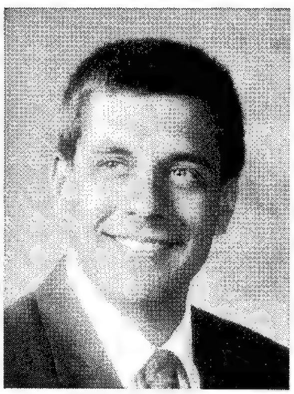
ANN E. TATE directs the Molecular Computing project in the Systems Research and Technology Department of NSWCDD. The focus of this project is the research of biomaterials and biomimetic architectures for advances in Navy computing applications. In addition to directing the project, Ann's principal research in the lab includes studies in protein biochemistry and molecular biology. The molecular biology research focuses on genetically altering proteins to promote attachment and patterning mechanisms for device fabrication using biomaterials.

Ann has been involved in molecular electronics and computing for the last five years. She is a founding member of the International Society for Molecular Electronics and Biocomputing. Her group has published over 20 papers in molecular computing and electronics in the last three years. Ann earned a B.S. in biology from the College of William and Mary in 1975 and a B.S. in mathematics from Mary Washington College in 1981. She has graduate credits in both math and engineering from George Washington University and Virginia Polytechnic Institute and is pursuing a doctorate in molecular biology.

During the early part of her 22-year career at NSWCDD, Ann worked on multiple combat systems engineering programs including guided projectiles, SEAFIRE, the Belknap Modernization Program, the DDG-51 Program, the Integrated Tactical Decision Aids Program, and RAIDS. Valuable operational experience was obtained while working on the Third Fleet and AEGIS tactical decision aids projects automating decision support on commercial computer systems integrated to Navy communications and links. In addition to this extensive experience in systems engineering and integration, systems design work in the AEGIS and TOMAHAWK programs has provided a valuable background for facilitating transition of biotechnology research applications to Navy systems.



JENNIFER L. BOYD earned a B.S. in biology (1991) from Mary Washington College. She then went on to receive an M.S. in applied molecular biology from the University of Maryland Graduate School, Baltimore (1993), through the NSWCDD graduate co-op program. She has been instrumental in NSWCDD's molecular computing program since the summer of 1990. Currently, Boyd serves as the head scientist in the Systems Research and Technology Department's Molecular and Cell Biology Lab, B507. She was the first Research Geneticist at NSWCDD to design and conduct research involving the identification and optimization of proteins through genetic engineering for use in molecular computing applications. Currently, her research focuses on the optimization and patterning of bacteriorhodopsin for computing applications.



DAVID W. CULLIN received his B.S. degree in chemistry from the University of Pittsburgh at Johnstown in 1984. He then went on to graduate school at The Ohio State University receiving his masters degree in physical chemistry in 1988. His masters thesis work involved measuring the angular distributions of sputtered atoms and molecules from metallic surfaces under ion bombardment. Dave received his Ph.D. in physical chemistry in 1991 working with Professor Terry Miller. His doctoral dissertation involved the molecular spectroscopy of organic free radicals in the gas phase using laser-induced fluorescence as a molecular probe. The bulk of his laboratory experience pertains to the interactions of light with matter including several types of molecular scattering spectroscopies. He joined the molecular computing project at NSWCDD in July of 1991. His current research interests are still in the area of molecular interactions with light. More specifically he is investigating real-time holographic biomaterials that can potentially be used in optical analog processing, memories, and materials characterization. He has also brought on line an ultrafast laser system that is currently being used to investigate, on a picosecond time scale, the fast photophysics of these biomaterials in an attempt to design faster optical processing devices.

ROBERT A. BRIZZOLARA is a Research Physicist in the Surface Evaluation Facility of the Naval Surface Warfare Center, Carderock Division. He received his B.A. (1981), M.S. (1984), and Ph.D. (1987) degrees in physics from the University of Delaware. He performed his graduate research under the direction of Professor C. B. Cooper on the effects of ion bombardment of surfaces and mechanisms of sputtered neutral cluster ejection. In 1988, he was awarded a National Research Council Research Associateship at the Naval Research Laboratory in Washington, D.C. There he performed research concerning sensor applications of the Scanning Tunneling Microscope. Dr. Brizzolara joined the staff at NSWC in 1990. His research interests include molecular electronics, nanotechnology, and surface science. In 1993 Dr. Brizzolara was selected as one of the NSWCDD Research and Technology Department's Young Professionals of the Year. Dr. Brizzolara is a member of the American Physical Society, American Vacuum Society, ASTM, and the International Society of Molecular Electronics and Biocomputing.

Data Visualization and Virtual Reality for Decision Support

Louis G. Batayte

This article discusses the concepts of data visualization and virtual reality (VR). It is meant to be a summary overview. While it is not all inclusive, it contains enough information to give the reader a basic understanding of the topic. It provides some examples of current uses for this capability and discusses some of the problems associated with this area. A brief philosophical discussion of how we got to this state follows.

According to Alvin Toffler, a widely recognized social thinker, human society is entering a third wave of evolution. The first wave occurred approximately 10,000 years ago when humans switched from a hunter gatherer society to an agricultural society. This reduced the time humans normally spent acquiring food and allowed them to engage in other endeavors. As society slowly progressed, a second wave occurred approximately 300 years ago when society entered the industrial revolution. This wave provided the mass production of machines that greatly enhanced the human race's physical capabilities. A third wave began approximately 50 years ago with the invention of the computer, a tool that rapidly processes information. This tool is in the beginning stages of producing a wave of greatly enhanced mental powers in human society.

As the power of computers increases, the complexity of the information they can process increases. As the complexity of the information being dealt with increases, the difficulty in understanding its meaning increases. Incorrect human decisions are often the result of misunderstood information.

An obvious challenge is to develop a variety of means to present this complex information to humans so that they can readily understand its meaning. Using computers to generate images, perform virtual reality, and produce data visualizations is a potential solution to understanding a wide variety of complex information.

Often, an important aspect of one's ability to extract information from an image is based on the color content of the image. Most of the figures in this article were originally generated in color even though they appear here as black and white images. The degree of information lost due to this transformation varies, depending on the reason color was used in the original figure.

Introduction

As little as 15 years ago, a computer was largely a corporate asset. Approximately 10 years ago, with the evolution of the general purpose microprocessor-based personal computer (PC), the computer began its rapid rise to what we have today—a personal information processing tool. Today, a personal computing device, costing between two and ten thousand dollars, has more computing power than a machine costing half a million dollars only 15 years ago. UNIX workstations

costing ten to fifty thousand dollars with supercomputer-like computing power and significant graphic capabilities are not uncommon in the workforce serving as a single individual's workstation. Groups of scientific workers share deskside UNIX super workstations costing one to two hundred thousand dollars. UNIX workstations, such as the Silicon Graphics, Inc., workstations, not only have state-of-the-art computing power, but also have state-of-the-art capabilities to convert data into pictures. This process of converting data into pictures is referred to as data visualization.

A team of researchers at the Naval Surface Warfare Center, Dahlgren Division's (NSWCDD's) Systems Research and Technology Department/Advanced Technology Group is currently working to develop data visualization techniques that exploit the power of computers to convert data into pictures. This team is supporting a variety of programs at NSWCDD, helping them to understand their complex data sets.

An often repeated phrase is "a picture is worth a thousand words." In computers, it may not only be worth a thousand words, but a million bytes of data. It is this process of visualizing data that must be developed. Compared to pictures, our brain processes text at a very slow rate. In another ten years, individual desktop computers will probably process billions of floating point operations per second, have gigabytes of memory, and terabytes, if not heptabytes of off-line storage. If the evolution of computer processing capability is to continue to be useful, we need effective means of presenting huge quantities of data to the user. These presentations must be developed such that the user can easily understand the meaning hidden in the data.

If a single picture is worth a thousand words, then a 30-second animation, at 30 frames per second, is worth approximately one million words. This process of displaying a series of pictures, in a sequential fashion, depicting how the data changes with regard to some other variable (e.g., time), produces yet another, more advanced way of converting data into information. By generating special displays, we can create stereographic pictures. This technique can be used to immerse the user in their data, yielding even more opportunity to uncover the hidden meaning in the data. We can also incorporate audio into the effort to further enhance understanding the data.

In the world of animation, an interesting aspect of pictures in motion needs to be discussed. Our eyes have some nerve cells that are not directly affected by light but are stimulated by the changing conditions of the light. They are, in effect, differential light sensors. If we display a series of pictures at a very slow rate, then these cells are not greatly affected and we only see the picture and its content. If we show the pictures at a faster rate, then these differential cells kick in and detect the changing nature of the pictures; i.e., they "see" the motion in the pictures. Thus, animations put added information into our brains. This information—the motion in the scene—transcends the actual content of the pictures. Even though we can't reach out and touch it, we can "see" this abstract quantity: motion.

Background

History

Most of the earlier techniques used to display data generated by computer were based on line drawings, such as those made by pen plotters. These drawings were generally generated during a post-processing phase that was, by modern standards, time-consuming and expensive.

Hewlett Packard, Inc., and Tektronix, Inc., achieved some of the first general-purpose cathode ray tube (CRT) graphics on the desktop with their microprocessor-based workstations. These workstations employed vector-refresh technology and created superb wire-frame, line-type drawings. When coupled with inexpensive desktop pen plotters and screen copy devices, they formed the basis for the first, personal data visualization systems. These workstation screens had a typical addressable resolution of one part in four thousand, and the lines produced were precise, crisp monochromatic vectors.

The next two major advances in graphics were the invention of raster display technology and the introduction of color. Raster display technology divides the screen into a two-

dimensional grid of square pixels. This idea converted the display screen into a digital device, with each pixel being a discrete entity. The process of creating a picture is now a matter of deciding what color to paint each pixel, thus creating a picture from a mosaic of single, colored squares.

Color

A short discussion of color might be useful to some. The human eye, in detecting light, happens to be made up of two general types of sensors: rods and cones. The rods are very sensitive to small quantities of light and are what give us our "night" vision. The cones are less sensitive and only useful during well-lit periods, e.g., daylight. The cones are subdivided into three subtypes: one is sensitive to light in the "red" region of the visible light spectrum, the second is sensitive to light in the "green" region, and the third is sensitive to light in the "blue" region. The rods are monochromatic and sensitive to light generally in the green region. When the rods are exposed to bright light, they effectively go into saturation, and shut down. When you are in a brightly lit room, you are seeing with your color cones and your rods are in saturation.

When you leave the brightly lit room and enter the dark of night, the cones shut down due to insufficient light, and the saturated rods require a recovery period before they can become effective, so you suffer temporary night blindness. This, of course, was a problem for many, including the Navy, operating at night. So they solved the problem by lighting the room with red lights in a wavelength far from the wavelength of the green-sensitive rods. Now, when you leave a room that is well lit with red light, where the red-sensitive cones are doing the visual work, and enter the dark night, the rods are ready immediately since they were not driven into saturation by the red light. An important feature of the red, green, and blue cones, is their overlap. The upper frequency limit of the sensitivity of the red cone overlaps the lower limit of the sensitivity of the green cone. This is important, since without this we would never see the color yellow, which is between red and green. When the color yellow is present, both the red and green cones are responding with signals to the brain, and the brain declares this must be yellow. This trivia is important, because it should now be apparent how it is possible to deceive the brain. By placing two lights, one red and the other green, close enough together such that the eyes cannot spatially separate the two sources, the brain senses both the red and green cones being stimulated, and declares the now apparently single light to be yellow.

Since the computer screen CRT is now merely a grid of squares, (i.e., pixels) each square can be subdivided into a grid of dots. Each dot is a different type of chemical phosphor. Different phosphors emit photons of light at different wavelengths when struck by electrons. By selecting three phosphors that emit one of red, green, or blue light, to stimulate the red, green, or blue eye cones, and by controlling the intensity of the red, green, blue emissions by controlling the electron flow impacting the phosphors, we can deceive the brain into believing there is just about any color on the screen we want. Just remember, there is no yellow or any other color on your computer screen or your television set, etc., other than varying intensities of closely spaced red, green, and blue (RGB) dots.

Mechanics

So let's draw a line on the screen. In the days of vector refresh, you merely located the two end points on the screen and swept an electron beam along the screen between these points, lighting up the monochromatic phosphors as you went, producing a crisp, clear straight line. On the raster refresh screen, we have to determine each of the pixels that will be intersected by the line and color those particular pixels the color of the line. This creates a line composed of a series of squares generally aligned with the line. Further, since the dimensions of the pixels are large compared to the thickness of a line, you get a line that suffers from what is commonly called the *jaggies*. This was the major complaint of people moving from vector refresh to raster scan technology—lines on a raster scan screen are definitely inferior to lines on a vector refresh screen.

Now let's draw a filled-in polygon. To make a long story short, there is no good, practical, general-purpose way to do this on a vector refresh screen. Vector refresh technology, in general, is only suited for wire-frame, line-type drawings.

On a raster scan screen, you merely determine which pixels are in the interior of the polygon and color them accordingly.

Finally, let's draw a complex scene on the raster scan screen. One way of doing this is using a *ray-traced* approach. Using this approach, we locate where the viewer is in the scene, and we place the pixel grid between the viewer and the scene. We then cast a mathematical ray, from the viewer's eye through the center of a pixel, and determine what objects in the scene this ray would collide with. Since the task is to determine a color for each of the pixels, if this is a simple scene containing all opaque objects, then we need to find which of the objects that the ray collided with is closest to the viewer. Since this object obscures whatever is behind it, this pixel will be the color of this closest object. We then cast rays through each and every pixel and determine each pixel's color. When we are done, the resulting mosaic will be a representation of the scene.

Since we are casting mathematical rays into a mathematical representation of the scene, we can include any effects we want. We can allow for translucent objects; which will mean the color of a pixel will be a combination of the colors of several objects along the ray. We can allow for reflection of light off shiny surfaces, so the color of a pixel is actually the reflected color of an object that is not hit directly by the ray but is hit by a ray ricocheting off the shiny surface from somewhere else in the scene. Based on the laws of physics as they relate to the propagation of light, we can produce shadows, glare, spotlights, etc. Modern graphics workstations use various approximation techniques to the full ray-tracing scene analysis to render scenes at a much higher rate than is currently possible using ray tracing. These graphics workstations use z-buffers, gouraud shading, phong shading, lighting approximations, texture maps, etc., to work their magic. But the folks making Star Wars type movies always use ray-traced images for their final super realistic pictures.

Images

An aspect of computing that needs to be discussed is that any information that is processed in a computer has to somehow be represented as a number inside the computer. Text in a computer is represented as numbers, with a standard accepted conversion table between the internal numeric value and the symbol that is displayed; i.e., the ASCII character set. In this environment of everyone doing their own thing, it is almost a miracle that the world got settled on the standard ASCII text conversion table. Once again, everything in a computer is numbers, and that is what the raster display technology did to the display screen, it converted it to a two-dimensional grid of numbers. This two-dimensional grid of numbers, when converted to RGB colors, produces a picture. This matrix of numbers is generally referred to as an image.

Since we have a two-dimensional matrix of numbers, we can manipulate this data using various mathematical algorithms. We can transform the first two-dimensional matrix into another, second matrix, convert the second matrix into colors, which of course creates a picture, and let our eyes appreciate the beauty of the transformation created by the mathematical manipulation of the numbers that represent colors. Of course, this matrix manipulation process is referred to as image processing. Some algorithms tend to reduce the degree of numeric change from one pixel to the next, and this produces a smoothed image. Other algorithms tend to enhance the numeric change and this can produce edge detection images. Of course if you create a matrix of numbers at some point you will want to store them in a file for later access. In this environment, with everyone believing they have a better way to store this matrix of numbers, we have evolved a sea of image file formats (e.g., tiff, giff, sgi, rla, etc.). Maybe one day the "perfect" format will emerge, and we will have a standard form for the exchange of image data, much like the ASCII text standard.

Hardware Approximations

For smooth animations, it is desirable to create and display 30 picture frames per second. The ray-traced approach based on physics is the correct way to render these scenes. Even though modern workstations can execute up to 50 million instructions in the 30 milliseconds between frames, they are still not fast enough to ray trace a complex scene at 30

frames per second. Thus, approximations to the ray-trace method must be used.

Once again, the problem is reduced to what color to paint a pixel, and that is generally the color of the object closest to the viewer. To determine what object is closest to the viewer, modern graphics engines use a z-buffer. A z-buffer is an array of memory with an entry for each pixel on the screen. This memory location will contain the depth value of the object that caused the pixel to have its current color. For example, consider drawing a solid, shaded triangle to the screen. This triangle will be defined by three vertices, each with a three-dimensional x,y,z value. If the screen is an x,y grid of pixels, then from the x,y values of the triangle vertices we can determine which pixels are aligned with the interior of the triangle. Thus at a given x,y pixel value, we can determine the x,y triangle value, and also determine the z value of the triangle surface at this x,y value.

Now we could change the color of the pixel, but first we will look in the z-buffer at the memory location that corresponds to this pixel location and examine the z value stored there. If the z value of our current triangle is closer to the viewer than the z value currently in the z-buffer, we will change the pixel color to the color of our triangle, and we will update the z value in the z-buffer to reflect the depth into the scene of the piece of the triangle that caused the color change of this pixel. If the value in the z-buffer is closer to the viewer than our triangle's z value, we will leave this pixel color and z-buffer value alone.

In the previous example, we changed the color of the pixel when the object we were trying to draw had a z value closer to the viewer. This was because the new object was opaque. If the new object was translucent, we could have blended the color of the new object with the color already at this pixel location. This process of blending colors is called alpha blending and is an approximation to rendering translucent objects. Note, when rendering opaque objects, the order they are rendered does not matter. The object closest to the viewer will win out in the end. When rendering translucent objects, the only way to get the correct final result is to render the objects in the correct order, from deepest in the scene to closest to the viewer. If an object is attempted to be rendered that is deeper into the scene than the current z-buffer value, it will be discarded and not blended in. In fact, if it is not blended in at the proper place in the stack of objects, it cannot be blended, since the pixel color has no history associated with it, just the current effective color.

The previous example assumed that the triangle object was a fixed color. Consider the example where one vertex of the triangle is red and another vertex is green. The desired effect would be to transition the color along the line connecting the vertices from red to green. This process of interpolating color is called gouraud shading and is another approximation that is imbedded in modern graphics workstation hardware. Note, the example of moving from red to green has a problem. As we transition from 100 percent red, 0 percent green to 100 percent green, 0 percent red; at the midpoint we would be at 50 percent red and 50 percent green. We might expect that half way between red and green is yellow, which is 100 percent red and 100 percent green. Gouraud shading will give you 50 percent red and 50 percent green. To properly use gouraud shading to shade colors from one hue to another, care must be taken to subdivide any polygons whose vertices color cross any of the primary, secondary, or black or white colors. The primary use of gouraud shading is to shade a given color to account for the effects of lighting.

If we look at a point on the surface of an object, under the influence of lighting, three categories of information are important:

1. The material the object is made of
2. The type of light
3. Three vectors

As for the vectors, the first is the unit vector from the viewed point on the object to the viewer's eye. The second is the unit vector from the viewed point on the object to the light source. The third is the unit vector that is normal to the surface of the object at the point being viewed. These, of course, are needed to compute the amount of light that is reflected from the light, off the object, to the viewer.

In general, there are three parameters that are used to describe attributes of both the light and the material. These are the attributes of ambient, diffuse, and specular.

Ambient lighting is a condition where the available light is everywhere. With this type of

lighting, none of the vectors influence or attenuate the color. The material definition generally has a value for the color of the object in the presence of ambient light. For the light, there is generally a color and strength of the ambient light emitted.

Diffuse lighting is an effect generated when light is reflected off a material like cloth. The light that reflects off the object is scattered in all directions. Thus, the attenuation in color is only a function of the light position vector and the object surface normal. The apparent color is viewer position independent. Once again, the color and strength of the diffuse light emitted is a light property, and the color of the material in the presence of diffuse light is a material property.

Specular light is the effect generated with light maintaining coherent reflection off a shiny object. This is the type of light that produces a sun glare off the window or chrome bumper of the car in front of you. This type of light is affected by a combination of all three vectors: light position, viewer location, and object surface normal. Once again, there is a definition of the color of the material in the presence of specular light. There is generally a further material parameter associated with the shininess of the material. For specular reflection, the reflection angle equals the incidence angle of the light striking the object. Thus, a viewer looking along this reflectance line will see the maximum specular light. The shininess parameter defines how rapidly the specular effect deteriorates as you move off this ideal viewing line. And, of course, there is a definition of the strength and color of the specular component of the light.

The final color of the object surface will be a combination of all these colored effects of light and material combinations. These computations are carried out in special hardware in a modern graphics workstation.

Referring back to our triangle, we now have a triangle in three-dimensional space with not only three x,y,z vertices values, but three i,j,k normal vectors. While our triangle happens to be a flat plate, and thus might seem to have only one surface normal, if it is a patch out of the side of a sphere, then we can compute a normal at each vertex, none of which equal the general normal of the plate. Now we throw this triangle into our graphics engine with the lights turned on, and the engine computes a color for each vertex. It then uses gouraud shading to interpolate colors for the interior of the triangle, producing a shaded triangle that simulates shading due to curvature. This is what is generally done in most graphics engines.

There is a problem with this method though. If this triangle had a specular reflection point in its interior, gouraud shading will never display it. There is another technique that interpolates the vertex normals, thus estimating normals for points in the interior of the triangle, and then uses this interpolated normal to compute a color. This is called phong shading, and it will do a better job finding specular points.

We discussed earlier that vector refresh terminals drew nice clean lines and raster scan screens drew lines that suffered from the jaggies. Well, a method to help smooth out jagged edges, either lines or the sides of polygons, is called antialiasing. This technique assigns some thickness to the line, computes what percentage of a pixel is actually covered by the line, and then blends into that pixel a percentage of the line color. This, in general, tends to blur the line and make it look better, especially when viewed from a distance.

Texture mapping is another technique worth mentioning that modern graphics workstations use to help approximate detail in a scene. Consider we have an image which, remember, is a two-dimensional matrix of numbers that, when displayed as colors, form a mosaic picture. Now consider we have a quadrilateral polygon. Consider we tell the graphics hardware to take the number and thus color in the lower left corner of our image and paste it to the lower left corner of our quadrilateral. We now paste the number/color in the lower right corner of the image to the lower right corner of our quadrilateral, and we will do the same for the upper left and upper right corners. Now we have a color at each of the vertices of the quadrilateral. Now, instead of using gouraud shading, we interpolate colors based on the image. For example, whatever color is halfway between the lower left and lower right corners of the image, put this color halfway between the lower left and lower right corners of our quadrilateral. The net effect of this interpolation of the image onto the quadrilateral is to paste the picture represented by the image onto a quadrilateral that we can translate, rotate, etc., in

three-dimensional space. This process of texture mapping is the current focus of the future of graphics workstation technology. This is a very effective way of producing what appears to be a very complex scene, which it would be if you were to actually build three-dimensional models of everything in the scene. The fact is, all you have done is paste a bunch of photographs on some simple polygons. But this is a very effective technique for creating visual simulations, such as driving a car around a town. All the pretty demonstrations you see on high-end workstations, such as Silicon Graphics workstations, are massive texture-mapped environments.

One final topic worth discussing is stereo viewing. One of the most desirable effects in presenting three-dimensional data to the human visual system is to convey the three-dimensional nature of the data. There are a variety of techniques to aid in this effort. One is the use of a perspective viewing transform. This transformation enlarges objects close to the viewer and reduces the size of objects farther from the viewer. A second technique is to shade objects with lighting. This shading of curved surfaces is an important visual clue to the brain. The last technique is to present a stereo pair of views. Using this technique, a separate view is presented to the left eye and the right eye. Each view is slightly different based on each eye having a slightly different viewing position in the scene. While the primary effect in creating a stereo view is the two different views, there are a variety of secondary effects that need to be considered.

Based on evolution, the human visual system has hundreds of million of years of development behind it. It performs tasks that most people don't even consider. For example, your brain knows when the muscles are pointing your eyes straight ahead. It also expects that if you are looking straight ahead, the muscles focusing your eye should be relaxed so you will be focusing at infinity. When most people try to look at a stereo pair of pictures that they hold in front of their eyes, even with a piece of cardboard blocking one eye from seeing the other eye's view, they cannot create the unified stereo image. The brain sees the two separate images, but it is looking straight ahead and focusing in close.

Clearly, this is not what happens in nature. If you are focusing in close, then you are looking inward. If you put the pictures in a viewer that has lenses that make your eyes focus at infinity to see them, even though they are close, then the combined stereo image instantly merges into a single stereo image. If the stereo pair of images is too large, the separation during presentation causes the eyes each to look outward, which of course never happens in nature, and thus you won't see stereo this way. Possibly, a viewer with lenses and prisms might be designed to view large stereo pair pictures. Another method that works for some people is to put the right eye view in front of the left eye and the left eye view in front of the right eye. The viewer now looks cross-eyed at the pair and sees a combined single stereo image. This works because the eyes are looking inward and focusing in close. Thus, the stereo image floats in the air where the view angles cross.

Data Visualization

Data visualization is the process of converting numeric information into meaningful pictures. In effect, it is the transformation of data from one form of representation to another.

In the 1960s and early 1970s, most scientists and engineers were happy to have access to a computer that could produce textual printed output. During this time, engineers would examine large quantities of data in tabular form, trying to both understand it and verify its accuracy. In the mid 1970s, things took a turn for the better. With the invention of the microprocessor and early desktop workstations, engineers acquired the ability to interactively generate simple, line, strip-chart-type plots of data. In many circles, these line graphs are still thought of as the only method available to examine data, even though modern graphics-type workstations have opened up many new avenues for depicting and exploring data.

Consider the following example from a recent weapon system test program. In this test program, several factors were being investigated. One part of the test involved crashing a semisolid object into a solid object. It was of interest to understand the dynamics of the reactions that occurred at the collision interface. The test was instrumented by placing four load-sensing instruments on the solid object where the collision would occur. Table 1 is a tabular sample of the

Table 1. *Load Data as a Function of Time*

Time	Load 1	Load 2	Load 3	Load 4
0.0000	560.0	504.0	672.0	728.0
0.0003	560.0	504.0	728.0	672.0
0.0005	560.0	560.0	728.0	728.0
0.0008	560.0	504.0	616.0	728.0
0.0010	560.0	560.0	784.0	728.0
0.0013	560.0	672.0	896.0	728.0
0.0015	616.0	1064.0	1512.0	784.0
0.0018	616.0	1848.0	2632.0	952.0
0.0020	728.0	3136.0	4536.0	1176.0
.	.	.	.	.
0.0497	1568.0	1568.0	2352.0	1848.0
0.0499	1512.0	1456.0	2408.0	1848.0
0.0502	1456.0	1400.0	2296.0	1792.0
0.0504	1344.0	1288.0	2128.0	1680.0
0.0507	1288.0	1232.0	2072.0	1624.0
0.0509	1288.0	1232.0	2016.0	1624.0
0.0512	1344.0	1232.0	2016.0	1624.0
0.0515	1344.0	1288.0	2072.0	1680.0
0.0517	1400.0	1288.0	2072.0	1680.0

data recorded during one impact. This is what I would refer to as a level 0 visualization. This, of course, was too much data to look at in tabular form and understand, and our visualization group was asked to help reduce the data by generating the typical line type, strip-chart plots. Overall, there were many tests, with many different parameters (not just the load data), and we generated many plots. However, the load plots generated looked like the ones shown in Figure 1. This is what I would refer to as a level 1 visualization. Actually Figure 1 is a composite of what were four separate plots: one for each load sensor, just plotted here on a common plot.

During the period of reducing the data, we were doing what we were tasked to do—the mass production and generation of the line plots. After the plotting task was finished, we happened to look at this load data and realized we had data that was correlated in magnitude, position, and time. We further realized we could develop another simple, animated display of this data. So we developed a "sponge rubber" cube (Figures 2 and 3). Each corner of one face of this cube would deform and change color according to the magnitude of the load data from a load sensor. The face was animated with the loads varying over time and the results observed. I would refer to this as a level 2 visualization. Viewing this data, we achieved a new and interesting appreciation of the dynamics of the data.

It also made some conclusions apparent. First, the quantity of data as represented by

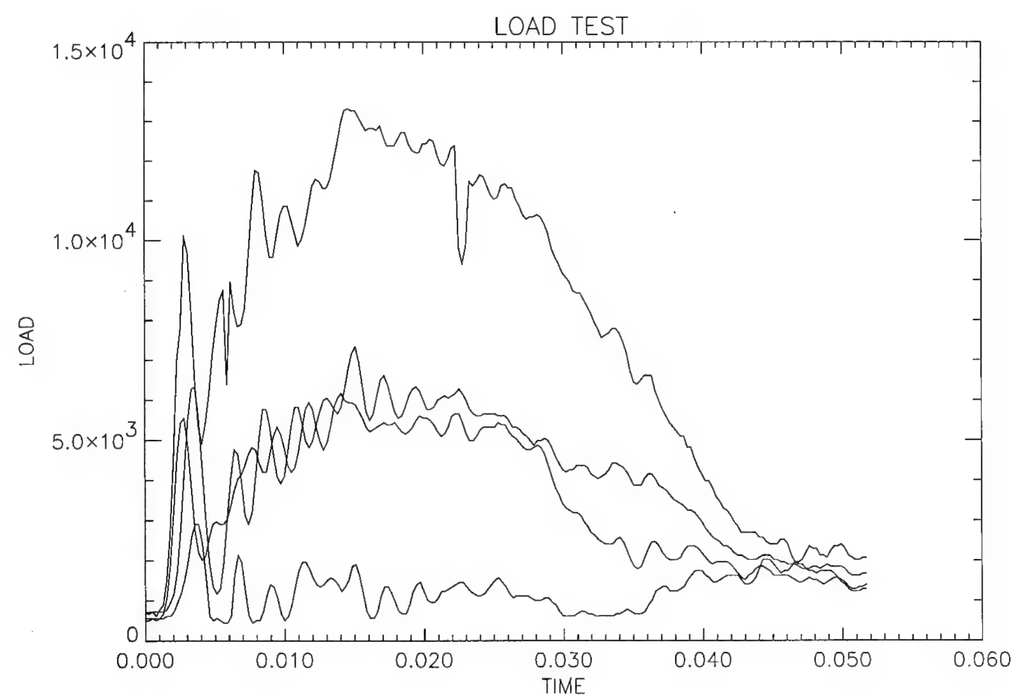


Figure 1. *Impact loading as a function of time.*

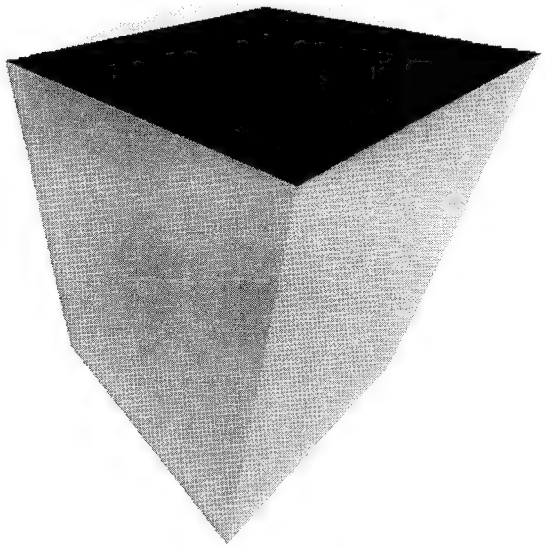


Figure 2. *Sponge cube with minimal load.*

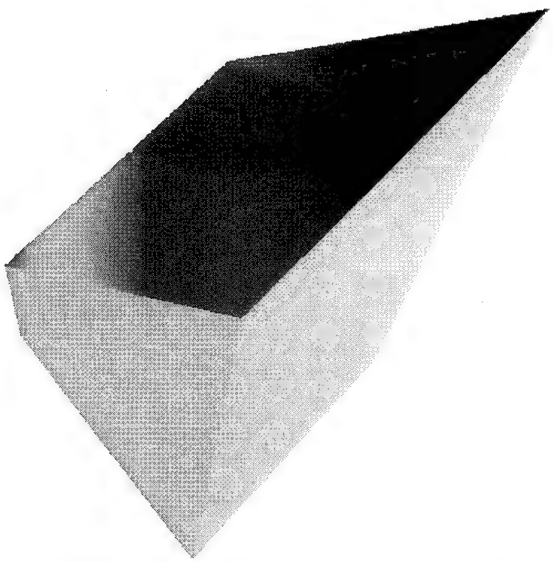


Figure 3. *Sponge cube with heavy load.*

Table 1 was overwhelming when viewed as a tabular level 0 visualization. Second, when converted to the level 1, line-type plots shown in Figure 1, the ability to understand the dynamics in the data was still uncertain. Further, if more load sensors were used, more plots were not going to help in a level 1 mode; it would only enhance the confusion. Third, when viewed as the level 2 animation, we had nowhere near enough data. We needed more load sensors in the test. If we had a matrix of 5 by 5 load sensors or maybe even 10 by 10 load sensors, we could have depicted the cube face as a grid and observed the full dynamics of the collision. We could have easily determined whether we had load spikes that were wandering around the face of the collision, like waves roaming around on the surface of a disturbed waterbed. We could have easily detected any grinding motion occurring during the collision, which was one of the original concerns.

Another fact that is obvious about level 2 type visualizations is that while they may be a wonderful tool to help scientists and engineers to understand data, it is impossible to adequately represent this information in a printed document such as this one, particularly when printed in black and white, not color. We have no ability to *see* the motion in a static printed picture, or even a series of static printed pictures. Maybe, one day we will have electronic printed paper that can store and display animated images.

The previous example of the "sponge rubber" cube, illustrates the need for scientists and engineers to keep current with the capabilities of computers to assist them in understanding their data. It also demonstrates one of the techniques of data visualization. That is, to take an abstract quantity (e.g., force) and represent it as a physical object, the cube face. In fact, there are two basic forms of data visualization. One form depicts how real physical objects behave, such as a re-created visualization of an aircraft in flight. The other is visualizing abstract quantities that have no real physical shape; e.g., temperature, pressure, force, etc. In modern graphics workstations, the technique used for creating pictures is to draw or render a picture of some geometric object. Thus, one additional task associated with visualizing abstract data is to determine some appropriate geometric representation for that abstract quantity; e.g., vectors depicted as three-dimensional arrows that may change in magnitude, direction, shape, and/or color.

Animations

Animations are an advanced form of data visualization. They are, however, a practical reality using modern high-performance graphics workstations. Today, animations may either be live, on the computer display screen or, in the traditional sense, recorded onto tape for off-line viewing. The traditional method of creating an animation is to create every frame of the animation separately and store each frame either to disk or directly to a single-frame editing tape recorder.

In the traditional method, an animation was not viewed until the tape-recorded version was played back. Today this technique is still used when the highest quality animation is required using ray-traced images. In general, however, working quality animations are producible, which run in real-world time, directly viewable on the workstation screen. Any animation is really a series of single-frame images presented to the eye fast enough to produce the illusion of continuous motion. Using a modern high-performance graphics workstation, it is possible to produce sophisticated animations at frame rates from 10 to 30 frames per second (in real-world time), with the computer creating each image by drawing each frame as it goes. The minimum usable rate to retain a sense of continuous motion is 10 frames per second with 30 frames per second being a more desirable rate.

To perform animations, there are two types of animation tools. One is an interactive tool that is used by researchers to explore their data to try to uncover the hidden meaning in that data. Using this type of tool, the researcher has complete control over the data display and can manipulate and animate the data in any manner necessary. When a researcher is using an interactive tool, quite often anyone looking on will be frustrated trying to watch. While the researcher in control sees one interesting aspect in the data and goes off to explore that element, the onlooker will have seen something else and want to go off in another direction, thus being frustrated not knowing where the researcher is going. Researchers exploring data this way can easily get captivated by what they are seeing, lose track of time, and ignore errors in the exploration process such as overshooting a viewing angle and backing up to get it right. It is this inability to study the display, get it right, and also sense time that makes it difficult to use an interactive tool to produce a tape recording that the researcher can take to show others.

A videotape made from an interactive session is generally hard to understand and probably boring for a general audience to watch. For this reason, a second type of animation tool is desirable. This is what is called a key-frame animation tool. Using this technique, a scripting file is created that describes where the viewer is, what the viewer is looking at, what time it is, etc. This script record is called a key frame. Each line in the file is another key frame telling where the viewer is next, what the viewer is looking at, what time it is, etc. In addition, there will be an indication of how many frames of animation are required for the computer to automatically generate as "in betweens" to transition from one key frame to the next. The computer animation tool will then use some type of interpolation scheme to estimate view location, etc., for each of the in-between frames. Thus, a complete script file will automatically generate a complete, precisely controlled animation sequence. This technique is very cumbersome to use to actually explore data, but it is the only effective way to create an animation presentation tape for viewing by others off-line.

There is an aspect of animations that should be noted. A problem can arise that results in temporal aliasing, or when objects appear to be moving in the wrong direction. This effect was quite often seen in old western movies when the wagon train was moving forward, but the wheels on the wagons were running in reverse. Of course this effect is generated by the stroboscopic effect of fixed-frame sampling times that are out of sync with the rotational rate. In fact, this effect can be a particular problem with rolling objects like projectiles. Since the purpose of data visualizations is to help people understand data, and not to confuse them, the areas in visualization that can cause problems must be understood by the people developing the visualization tools, and adequate allowances need to be made to eliminate any confusing or erroneous artifacts.

Virtual Reality: the Basics

The current, popular concept of VR began in the late 1980s. Certainly, the concept was conceived much earlier, but the technology and funding from National Aeronautics and Space Administration (NASA) came together in the mid to late 1980s when NASA developed some of the early Head-Mounted Displays (HMD). The basic concept was to immerse the user into a stereo three-dimensional interactive environment controlled by a computer. The HMD has a separate screen for the left eye and the right eye. The computer displays the left and right eye views of the environment on these screens. Further, devices are attached to the HMD to track the

movement of the head and, thus, the scene displayed could be automatically altered by the computer as the head is moved. With the addition of three-dimensional hand-controlled devices, the hands could be used to provide additional input to the computer to control the movement of the user in the virtual environment. Thus, the ability to immerse a user in a virtual, computer-controlled environment was born.

As the news of this capability spread, more and more people's imaginations were stimulated to the possibilities that might be obtained using this technology. By the early 1990s, as this turned into the latest fad, more and more companies saw marketing potential associated with this area, if only their products could be associated with this latest craze. Thus, the definition of VR began to expand, to accommodate more and more players. Today, the concept of VR has expanded to include almost any means of presenting realistic, real-time interactive, computer-generated scenes in front of a user.

There are three variable elements associated with the VR environments:

- The amount of detail in the scene, which is generally limited by the number of pixels on the display surface
- The amount of immersion into the scene, which is generally a function of the type of stereo viewing system employed
- The amount of interaction with the data, which is a function of application, the data itself, and the control devices available to the user

Depending on the application, each of these variables will assume a degree of importance. Consider an event reconstruction effort. The level of detail may be very important to adequately understand the event. The interaction with the environment will probably be of low importance since you want to re-create what happened, not allow the user to alter that event. Finally, the need for immersion into the data will generally be a function of the data itself.

Event Reconstruction VR

Scientists and engineers in the Advanced Technology Group have developed a capability to produce event reconstruction animations. The data to drive these animations can be either from simulations or actual data acquired during a field test. To date, the Data Visualization/VR team in the Advanced Technology Group has generated several animations in support of a variety of programs at NSWCDD. They are using high-end Silicon Graphics, Inc., graphics workstations with both in-house developed software and commercial software, such as Wavefront. They have video recording capabilities with a single-frame BetaCam recorder and an editing VHS/SVHS recorder. In addition, they have printing capability to a Canon color laser printer/scanner and a Codonics thermal dye sublimation printer.

One example of an event reconstruction was the flight tests of the Standard Missile Leap Program. There were four flight tests, FTV-1 through FTV-4. Animations of the telemetry data acquired from FTV-1 and FTV-2 were generated by the Advanced Technology Group as part of NSWCDD's general support for the flight-test effort. These first two animations were so well received by the Standard Missile Leap Program Office that they tasked NSWCDD to provide the animations for FTV-3 and FTV-4. These animations, since they were based on the actual telemetry data from the test, were to be produced as a quick-response effort within a matter of a day or two following the test. Using the data acquired from the telemetry systems aboard both the Leap target vehicle and the Standard Missile Leap test vehicle, it was possible to re-create the test from any camera position desired. The trajectories of the vehicles were depicted as lines in three-dimensional space so that long-range overall views of the trajectories could be seen (Figure 4). Cameras close to either the target or test vehicle

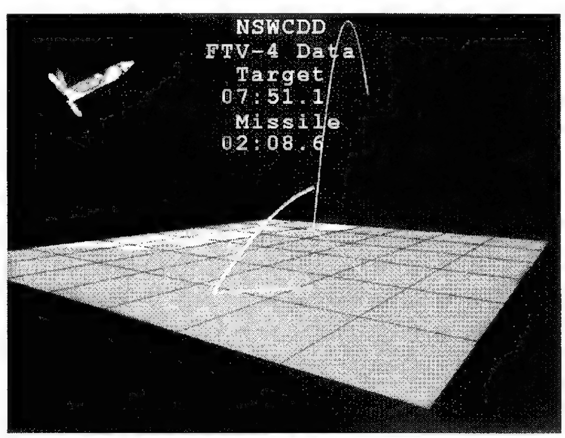


Figure 4. *FTV-4 overall trajectory with thumbnail view of third stage.*

could be used to observe the body attitudes and dynamics of the vehicles in flight, including staging separations, thruster firings, and rocket motor firings (Figures 5 and 6).

Any data recorded from the telemetry system can be depicted in some form or another. This type of animation gives the viewer an immediate understanding of the event from a high-level, natural perspective. The effect is one of the viewer actually observing the event. The effect of observing the dynamics of the event are impossible to achieve by any means other than an animated presentation. Also, by having this high-level understanding of what transpired, the typical, more detailed explanations of particular anomalies using static viewgraphs and the standard strip-chart-type data plots are much easier to understand.

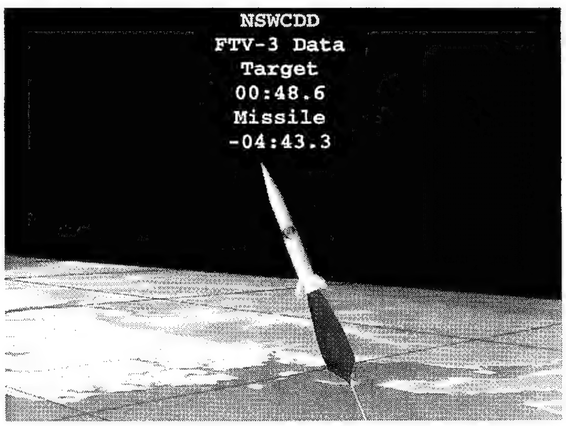


Figure 5. *FTV-3 target vehicle in flight.*

To reiterate the point once again, the purpose of any data presentation is to provide understanding of the data, either to the people involved in the effort or to other less knowledgeable but interested audiences. To this end, all appropriate methods of presenting the data—i.e., printed numeric values, strip chart line drawings, and animated displays—must be used. It is only in the last several years that the power of computers and computer graphics has evolved to allow practical, general purpose use of the animated data displays.

Another example of event reconstruction VR developed by the Advanced Technology Group is the animation of a simulation of a mathematical model of a human pilot attempting to fly a damaged aircraft. In the world of aircraft vulnerability, the ability to predict how much damage

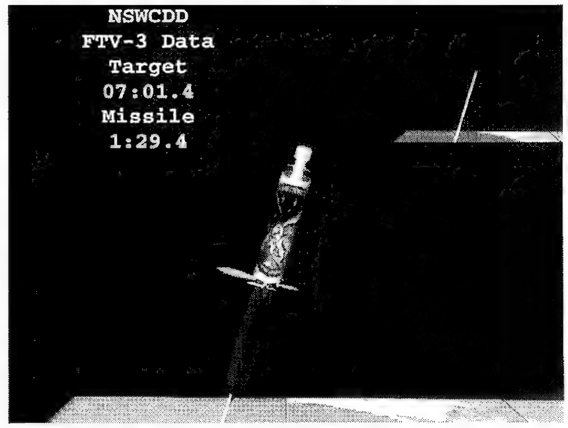


Figure 6. *FTV-3 Standard Missile Leap Third Stage during boost phase.*

must be done to an enemy aircraft in order to defeat it has been a difficult task due to the uncertainties surrounding the ability of a human pilot to compensate for the damage. The scientists and engineers in the Lethality and Weapons Effect Branch developed a mathematical model of a human pilot, trained this model to fly a simulated undamaged aircraft, then let it attempt to fly the aircraft with various types of damage. When they thought they had it flying correctly, they asked the Advanced Technology Group team to create an animation of the plane being flown according to the output of the simulation. We developed the animation and adjusted the frame rate of the animation so that one second of actual wall clock time re-created one second of simulated flight data. Thus, the images on the screen were a real-time view of the dynamics of the aircraft. The initial animations were of the undamaged aircraft performing various maneuvers.

The initial reaction of the engineers viewing their plane on the screen performing real-time maneuvers was, "Why does it look like the pilot is flying an airliner, not a high-performance jet fighter?" They went back and reviewed the constraints on their pilot and concluded the gains in the equations needed to be adjusted to give the pilot much more capability with this aircraft. The reality of the sluggish performance was not obvious looking at the strip chart plots of the data they had been producing. It became apparent only when they were able to view the actual animated performance of the aircraft. When these engineers were looking at their strip chart plots, they were mentally trying to create images in their heads of what the plane was doing. This, of

course, is potentially biased by their preconceived ideas of what they think it is doing. When the computer animates the data, there are no preconceived biases; it shows the viewer exactly what the data says it is doing. This surprise at what the data is actually doing as opposed to what the brain imagines it is doing based on strip chart plots, has occurred on several occasions, with many skilled scientists and engineers.

One of the damaged tests with this mathematical manned model was to remove part of the starboard wing and stabilator (Figures 7 and 8). When the results of this simulation were animated, an interesting thing happened. With part of the starboard wing removed, there would be a loss of lift force on that side of the plane, so the stronger lift on the port wing should cause a moment imbalance, causing the plane to roll, with the port wing rising and the starboard wing falling. Instead, the port wing fell, while the starboard wing rose. Upon investigation of this apparent error, it was determined that the added loss of the stabilator had caused a loss of the moment, keeping the nose up and, thus, a positive angle of attack. Before the plane could respond to the roll moment imbalance, it responded to the pitch moment imbalance; the nose dropped, and the angle of attack went negative. This caused a load reversal on the wings, which produced the observed roll condition. The original angle of attack was small, and this change from slightly positive to slightly negative was not noticed in the original strip chart data plots. Also, the sign of the roll angle was not noted as significant in the strip chart data, so the whole phenomenon went unnoticed until the animation made it very apparent.

Other uses of animations for event reconstruction have been used by the engineers in the Guns and Munitions Division to demonstrate the performance of their proposed guided munitions concepts. We have developed a family of similar animations depicting the Terminal Defense Round (TDR), the Rockwell International, Inc., SCRAMSHELL™ concept, and an Extended Range Guided Munitions (ERGM) concept (Figures 9 through 11). All of these are animated recreations of six-degree-of-freedom simulations of the concept rounds. With each of these, five to ten minutes of narrated animation can provide an audience with an understanding of the concept that an hour of viewgraph presentations may fail to achieve. In fact, the most effective presentation is one where the animation is used to give the audience a high-level understanding of the overall concept, and the details are filled in with the traditional viewgraph-type presentation.

A final example is an animated reconstruction of a simulated biological attack on the city of

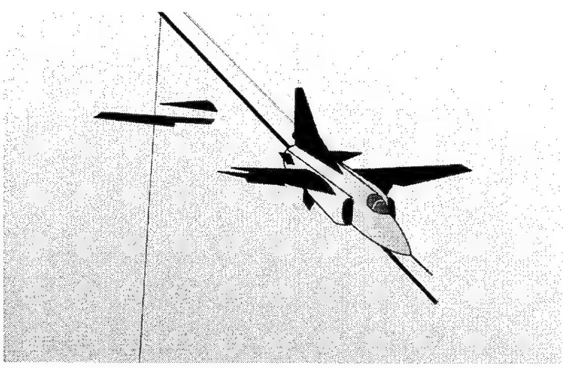


Figure 7. *Damaged aircraft wing section and horizontal tail removed at time approximately equal to time of damage.*

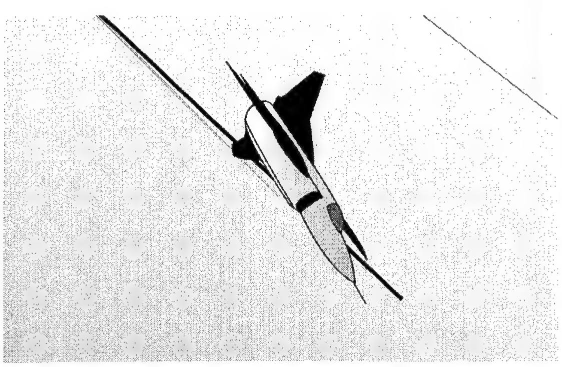


Figure 8. *Damaged aircraft wing section and horizontal tail removed at time equal to one second after damage.*

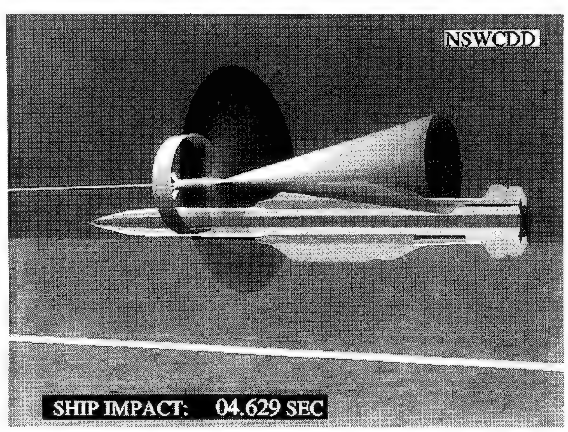


Figure 9. *Terminal Defense Round (TDR) concept at point of intercept showing TDR fuze cones and expanded ring of warhead fragments.*

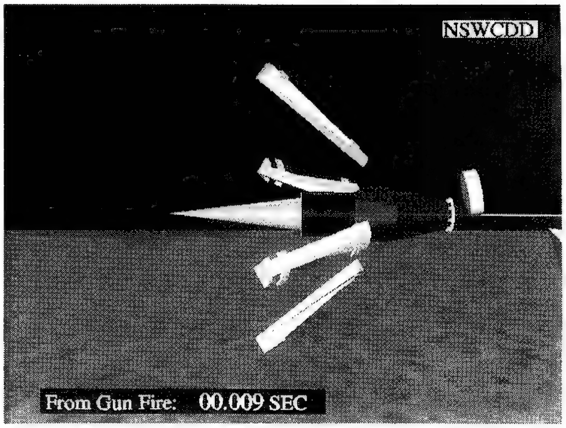


Figure 10. *Rockwell International Inc., SCRAMSHELL™ concept with motor ignited and sabots separating.*

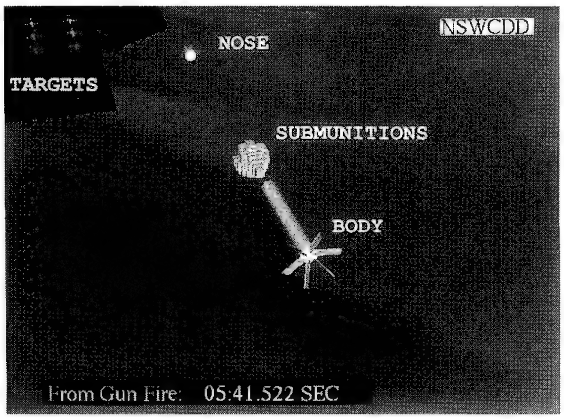


Figure 11. *Extended range guided munitions (ERGM) concept approaching target area with nose separated and submunitions ejected.*

Riyadh in Saudi Arabia. The scientists and engineers in the Chemical-Biological Systems Analysis Branch are concerned with the ability of the Navy and other services to protect their troops against chemical and biological warfare (CBW). They have developed simulations to predict the effects of a variety of chemical and biological agents that might be used against our troops and civilians during a war. These simulations were used by the scientists and engineers of the Chemical-Biological Systems Analysis Branch as part of the Global '95 war games conducted at the Naval War College (NWC) in Newport, Rhode Island. One of the simulated war events took place in the Middle East, and a coordinated biological warfare (BW) attack by a team of terrorists on the city of Riyadh was part of that exercise. Several teams of terrorists were simulated to release agents upwind of the city, so that it would be carried over the city. The resulting cloud and its propagation over time was simulated and subsequently animated (Figure 12). The effects of this cloud on the inhabitants of the city were predicted. From watching this animation it is clear why CBW weapons are termed weapons of mass destruction. The predicted result was that a very large percentage of the city's occupants were killed, and that there was very significant impact on a large area of the surrounding countryside. This animation also demonstrates the limitations of the current computing and computer graphics capabilities. There is such a large quantity of three-dimensional, cloud-type data generated by a simulation of this type that today's computers are not capable of handling it in a timely, interactive fashion. Approximations and simplifications to the data have to be made to produce animations that run at interactive speeds.

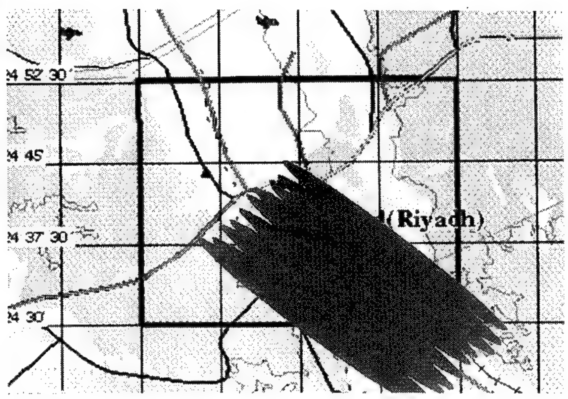


Figure 12. *Single-frame snapshot from an animated simulation of biological clouds drifting on the wind, overlaying a map of the Riyadh area.*

Immersive VR

The Advanced Technology Group entry into the immersive VR environments is currently evolving. The earliest capability we obtained was Crystal Eyes shuttered liquid crystal display stereo glasses. These glasses, when combined with an infrared emitter and a computer monitor operating at 120 Hz interlaced, are capable of creating the illusion of a stereographic image floating in space in the vicinity of the computer screen. Recalling that the purpose of computer-generated images is to present data or information in a manner that promotes enhanced understanding

of that data, it is then this author's opinion that this Crystal Eyes type technology, in combination with computer screens, has limited usefulness in the type of visualizations required at NSWCDD. In the area of three-dimensional shaded images, there is little or no enhancement of the viewer's appreciation of the animation or visualization. The one area where this, or any stereo presentation, greatly enhances the viewer's appreciation of the scene is in data displayed as either a wire frame object or a cloud of points. In both of these environments, it can be very confusing to try and determine which line is in front of what other line, or which point is in front of another in a monoscopic display. The shading used in solid images is not there as a visual clue, and the change in the size with depth is generally not present, with the line or point sizes being fixed. Thus, in these environments, the stereo glasses' approach to enhancing the three-dimensional nature of the data can be very helpful.

An enhancement of this stereo glasses' approach is to project the image on a full wall screen. When this is done, the size of the image becomes large compared to the viewer size, and this technique does enhance the quality of the presentation. This holds for three-dimensional shaded images as well as wire frame and point clouds. A monoscopic example of this technique, taken to an extreme, is the IMAX projections of movie film on fifty-foot-tall screens such as those at the Smithsonian Institution's Air and Space Museum in Washington, D.C.

As yet another iteration of the stereo glasses' approach to enhancing the visual sensation is the CAVE concept. A CAVE is a room with preferably three walls and a floor. Each of these is actually a projection screen. In this room, the viewer has on the stereo glasses and a head tracker. The views projected on the screens, walls, and floor are the solid-angle slices of the viewing volume surrounding the eye point in the virtual computer environment. This technique puts the viewer in an environment where the viewer is effectively looking at the world through a pair of sunglasses. The scientists and engineers in the Advanced Technology Group are currently beginning their investigations of this technology.

The final technology to produce immersion is to create a device to place a separate scene in front of each eye. There are two similar technologies providing this capability.

One is the stereo BOOM technology. This device is a box with eyepieces, much like binocular eyepieces. Inside the box are two small television screens, one in front of each eye. The box is supported on a boom structure to counterbalance the weight of the box with its electronics. In using this device, the user looks into the box and can move around while holding on to a handle on the box. The boom is articulated and allows a limited amount of translation and rotation of the viewing device. The disadvantage of this device is that it is somewhat unnatural to use when compared with everyday devices. An analog to this device is the periscope on a submarine. Of course, a periscope can rotate only around; the boom can translate and rotate about three axes. The advantage of this device is that the high-resolution screens can be placed in-line, with the eyes giving the user direct view capability. At the same time, the boom counterbalances the weight.

The other device for presenting individual images to the viewer is the HMD. This device is a helmet that is placed over the viewer's head. It is generally equipped with a head tracking device, such as a magnetic sensor. The most common version of this HMD uses small LCD direct-view screens. This limits the resolution of what the viewer sees and, thus, the detail. The other variation is an HMD with a high-resolution, one-inch diagonal screen monitor. To balance the weight of this device, the monitors are mounted on the sides, and lenses and beam splitters bring the image to the viewer's eye. This has the advantage of high-resolution images, but also has the disadvantages associated with the beam-splitter device.

The Advanced Technology Group has an HMD system. It consists of a high-resolution HMD from n-Vision, Inc.; two RGB Spectrum, Inc., parallel-to-serial signal converters; two SGI second-generation, high-end graphics workstations, with a third low-end SGI system to act as a controller; and an Ascension, Inc., Flock of Birds magnetic tracking system, a head tracker, and a three-dimensional flying mouse. The SGIs are networked together to share data. This approach to sharing data is too slow, and plans are underway to acquire some global shared memory, e.g., scramnet.

This equipment has only recently been acquired and is being used in its initial test in support of the Combat Systems Advanced Concepts Technology Laboratory (CSACT) at NSWCDD. CSACT is developing a laboratory environment to investigate advanced Combat Information Control (CIC) rooms. The first CIC will be constructed of wooden mockups. The Advanced Technology Group will be duplicating this mockup environment in the HMD-based VR environment (Figures 13 and 14). With both the VR version and the wooden mockup version, researchers will begin to get a feel for the types of problems relating to CIC development that can be explored using modern HMD VR environments. This can be directly compared to the mockup environment and evaluations made. The ultimate desire is to be able to create real environments, such as CICs, that work the first time when they are built. All the prototyping and redesign has been done in the most cost-effective ways. It is believed that the first phase of weeding out problems can be done in a very cost-effective manner using HMD VR environments. In this VR environment, CICs can literally go from the drawing board to the evaluator in an instant.

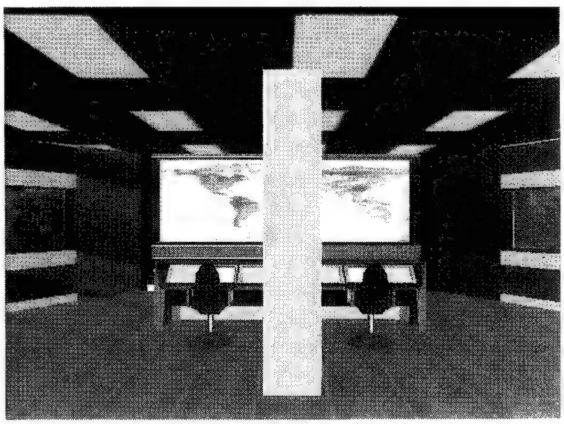


Figure 13. *CSACT laboratory in virtual reality space—view looking toward front of room (ceiling support post in the middle of view).*

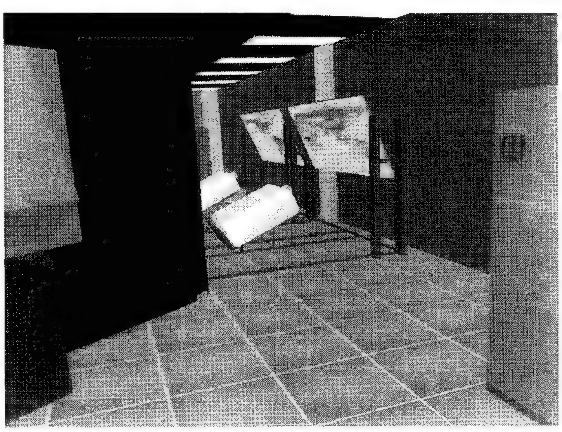


Figure 14. *CSACT laboratory in virtual reality space—view behind the set, showing the big-screen, rear-projection system.*

Deterministic Images

This final section is a little food for thought. In one of the earlier discussions it was pointed out that an image is a two-dimensional matrix of numbers. When these numbers are interpreted as colors and displayed, a grid of pixel colors appears that our brain combines into a single picture. There is an inverse of this process that involves scanning a picture into the computer. During this process, the user can select the number of pixels in the horizontal and vertical dimensions and, by scanning the image, convert it to a two-dimensional matrix of numbers. Since all we are looking at is a two-dimensional matrix of numbers, we could fill this matrix in any manner we choose. One such method is represented in Figure 15.

The dimension of this screen matrix was set at 640 by 480. In pixels, this approximates the resolution of the typical NTSC television picture. Since this computer program is a simple loop over a constrained value, it has a finite life and will eventually terminate. This, of course, means it will produce only a finite number of images at the dimensioned resolution. But it will also produce, somewhere in its loop, every matrix of numbers anyone can create by scanning in a picture at this same resolution and 24-bit color. If we were to do a frame grab from a television show, at the 640 by 480 NTSC pixel resolution, 24-bit color, we would create a matrix of numbers that this program would also eventually create. The implication is that this program would create all the frames of all the television shows that were ever shown. It would also create all the frames of all the television shows that ever will be shown. At the resolution of a television picture, it will create the portraits of all the people that have ever lived and the portraits of all the people who ever will live. It will create images of the Big Bang,

life on all the planets everywhere in the universe, and anything else you care to see.

There is a saying that an infinite number of monkeys, given an infinite number of typewriters, and an infinite amount of time, will create all the works of Shakespeare. This program will create the equivalent of those scanned page images, at all magnifications, somewhere in the series. Unlike the infinite number of monkeys, this deterministic series is not infinite; it's a finite series.

The number of images this program will create is undeniably a large number. Even generating a million pictures per second, there has not been enough time since the Big Bang to exhaust the loop. I haven't checked with

```
/*
 * THIS PROGRAM CREATES ALL THE PICTURES IN THE UNIVERSE
 * FOR ALL TIME - PAST, PRESENT, AND FUTURE
 * AT THE GIVEN RESOLUTION   - by LOUIS BATAYTE
 */
/************ The Global Definition Stuff *****************/
/**/
/* #define MAX_PIXEL_VALUE            1    - black/white picture */
/* #define MAX_PIXEL_VALUE          255   - 256 color or grey */
/* #define MAX_PIXEL_VALUE    16777215   - 24 bit RGB picture */
/* #define NUMBER_OF_COLUMNS       640   - TV resolution  */
/* #define NUMBER_OF_ROWS            480  - TV resolution  */

#define MAX_PIXEL_VALUE    16777215
#define NUMBER_OF_COLUMNS       640
#define NUMBER_OF_ROWS           480

#define  MAX_PIXELS    NUMBER_OF_COLUMNS*NUMBER_OF_ROWS

long screen[MAX_PIXELS]; /* initialize to all 0 if necessary */

long bump(long);             /* prototype for bump procedure */
void show_screen(void);  /* prototype for show_screen procedure */



/********************** The MAIN Program **************/
main()
{
        /* one line main program */
    show_screen();while(bump(0))show_screen();
}

/************ The Matrix Iteration Procedure *************/
long bump( long i )
{
   /* recursive four line procedure to increment screen matrix*/
  if( i > MAX_PIXELS )  return 0;  /* we are done!!!!!! */
  else {if(screen[i] >= MAX_PIXEL_VALUE )
          {screen[i] = 0; return bump( i +1);} /* overflow, bump next */
        else {screen[i]++; return 1; }    /* bump it and return success */
  }
}

/*************** The Screen Display Procedure *************/
/************** An Exercise Left To The Student *************/
void show_screen()
{
/* DO WHATEVER IS NECESSARY TO DISPLAY THE IMAGE
    STORED IN MATRIX  screen[]  */
}
/****************** End Of C Program *****************/
```

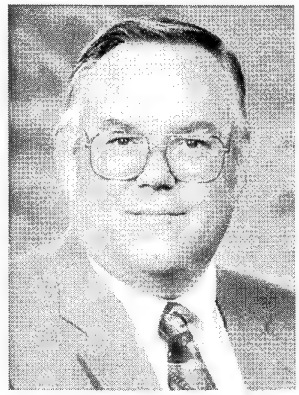
Figure 15. *Simple recursive computer program written in C.*

Stephen W. Hawking, so I don't know if there is enough time left—until the end of time—to finish the loop, even if we started today. And yet, if you imagine carrying a camcorder around the whole universe for all time, what would seem to be an infinite number of possible pictures will all be created within this finite set.

Summary

We are entering into the third wave of social evolution and are experiencing an explosion in the quantity of data available to us. If we fail to understand the data available to us, we will probably make poor decisions regarding that data. To this end, we must do whatever is necessary to uncover the hidden meaning in the data. One of the tools available to us is the power of our visual eye, brain connection. We, as humans, can extract significant amounts of information from pictures. Thus, we must use the power of computers to convert data into pictures so we can easily and rapidly absorb the information inherent in the data. Various programs at NSWCDD are embarking on this road to understand their data, and the scientists and engineers in the Advanced Technology Group are working to support their needs through the use of innovative data visualization and VR techniques.

The Author



LOUIS G. BATAYTE received a B.S. in mechanical engineering from the University of Connecticut in 1967 and has been employed at NSWCDD since that time. He has extensive experience in the Air Target Vulnerability Group in the areas of missile flight simulation for determining missile kill criteria, warhead fragment kill methodology, and target description modeling. In the early 1980s, he was involved in the Guided Projectile TECHEVAL and OPEVAL performing data extraction, reduction, and display. He developed a real-time data acquisition and display system and obtained test experience both on land and at sea during this program. In 1987, he moved to the Scientific and Computing Systems Division where he began his full-time commitment to the area of data visualization as the leader of their graphics efforts. In this position, he also led a successful effort to obtain a Division-wide major graphics workstation contract. He is currently employed in the Advanced Technology Group of the Systems Research and Technology Department leading their efforts to support various programs at NSWCDD in the areas of Data Visualization and Virtual Reality.

Promising Gains Made in Mammography Thanks to Dual-Use of Military Technology in Statistical Analysis and Image Processing

Richard A. Lorey, Jeffrey L. Solka, George W. Rogers, David J. Marchette, and Carey E. Priebe

Research begun for target identification by pattern recognition has been applied to mammographic computer-assisted diagnosis. Using computational statistics, feature extraction based on fractals and incorporating segmentation boundaries led to probability density estimation and classification based on discriminant analysis. The results of applying these techniques to mammography are very promising and are reported herein. These include possibilities of more thorough screening and earlier anomaly detection. The promising results of these limited mammographic studies are discussed in their own light and in comparison with others' work.

Introduction

Locating and identifying potential targets has historically posed problems in warfighting scenarios. As evident in Desert Storm, modern warfare's rapid deployment, quick-strike capability, and smart weapons usage has worsened this situation. Future warmaking capabilities' need for faster, or even on-the-fly, mission planning will exceed the limits of today's target-identification technology. It is in this light that the research described here was sponsored by the Office of Naval Research and was undertaken by the Naval Surface Warfare Center, Dahlgren Division (NSWCDD).

Initially, we consider the problem of discriminating between different classes of objects in a remote sensing scenario. In particular, we investigate the problem of classifying the different types of scenery in an image like the one depicted in Figure 1. Our goal is to determine how well we can distinguish, say, the buildings in this image from the rest of the image.

This investigation takes place at the local level–that is, we consider feature vector observations relating only to an individual pixel (or more correctly, to an individual pixel's local neighborhood) and to the texture of the image in that locality. Thus, we will not consider morphological features that may be obtained by segmentating the image.

Texture-related information is extracted using the "covering method," which is a fractal-based technique used on a greyscale image such as that in Figure 1. This approach to feature extraction and its application to image processing problems has been studied extensively.[1-4]

The features obtained are based on Richardson's power law. At each pixel, a local "area" estimate of the greyscale image is obtained for a number of different-sized neighborhoods, or scales. A regression of log (scale) versus log (area) yields three obvious features:

- The slope of the regression line (Feature 1)
- The y-intercept of the regression line (Feature 2)
- The significance of the regression hypothesis–the logarithm of the F-test (Feature 3)

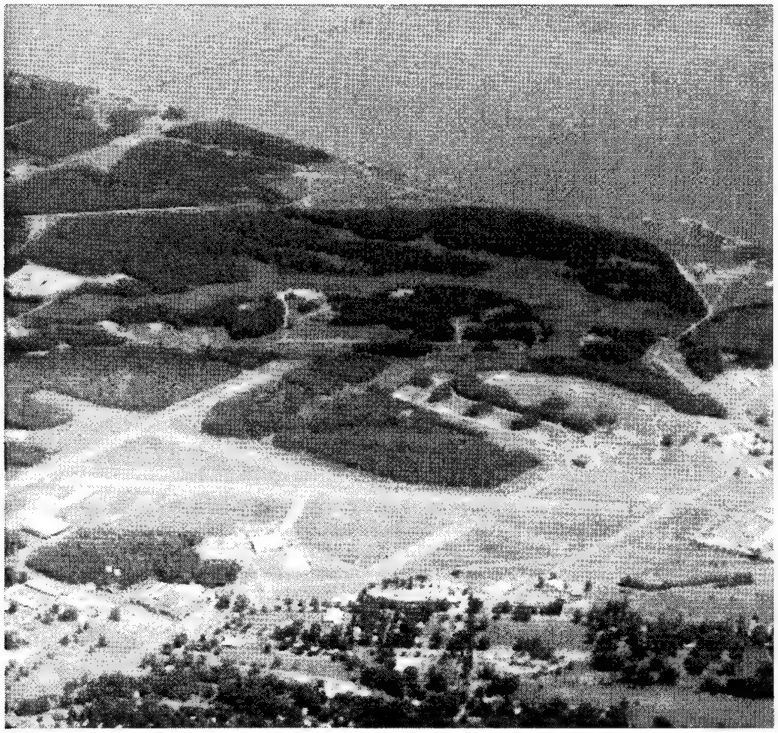


Figure 1. *Gray-scale aerial image of NSWCDD, Dahlgren, Virginia.*

We consider all three of these features to some extent but, for ease of presentation, we occasionally restrict ourselves to fewer dimensions.

Using the methods described below, probability density functions for the class "building" and the class "nonbuilding" are obtained (Figure 2) and a discriminant boundary determined. Figure 3 depicts the discrimination results at a particular likelihood level. This figure indicates that the areas classified as "building" (the areas covered by black overlay) almost always contain a building, and that the false alarms (black-overlaid regions containing no building) are quite few in number. A thorough discussion of this work and generalization performance is found in Priebe et al.[5]

The technology necessary to distinguish between manmade and natural objects can also be used to identify any class of object. Thus, with the advent of "Technology Transfer" it was natural for NSWCDD to apply its military expertise to other areas. The Research Triangle Institute and the Federal Laboratory Consortium Demonstration Project on Critical Industry Needs

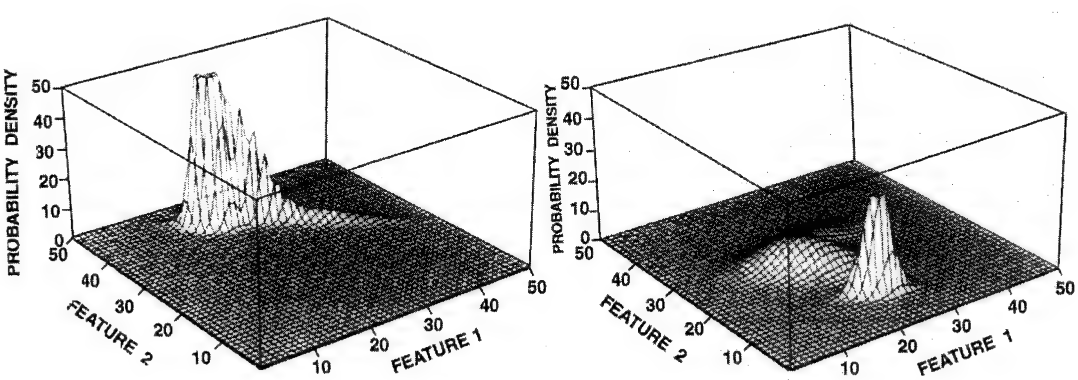


Figure 2. *Adaptive mixtures of probability density estimates for the classes "building" and "nonbuilding."*

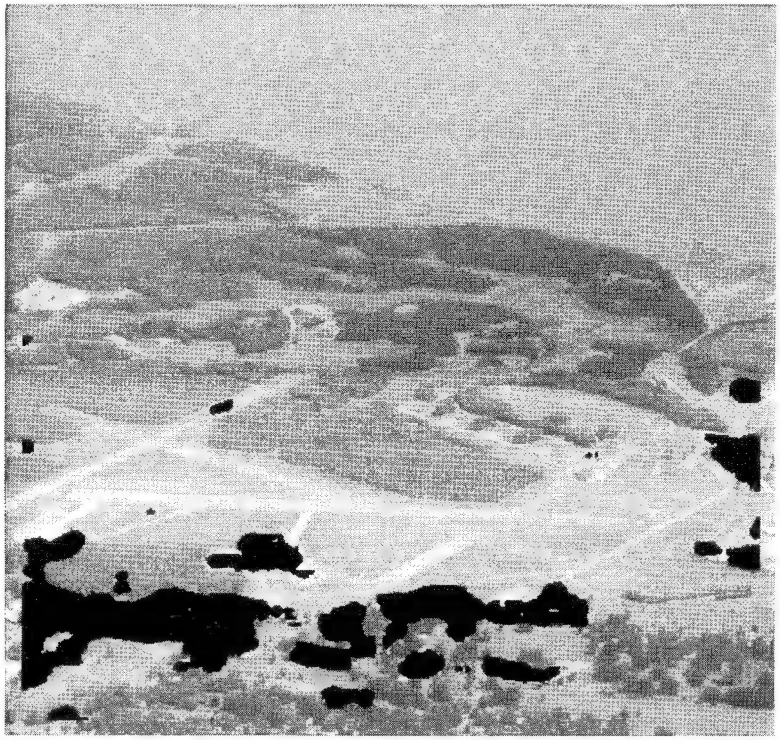


Figure 3. *Discrimination performance on entire image. Black overlay indicates pixels classified as "building."*

opened the door for the application discussed here. Specifically, an August 1992 National Cancer Institute problem statement called for software for Computer-Assisted Diagnosis (CAD), image processing, and pattern recognition for use in digital mammography systems. It is in this light that our research has been directed towards applying this technology to mammographic CAD. Successful efforts in this endeavor—potentially solving a more difficult pattern recognition problem—could well result in further state-of-the-art advances.

Computational Statistics Pattern Recognition

Our method for solving the pattern recognition problem uses computational statistics.[6] This theory involves very large data sets and does not incorporate assumptions about the parametric behavior of the data. Seemingly intractable problems can sometimes yield to these techniques.

Again, we consider greyscale digital images, with each class of object in the image characterized by a pattern or texture. Image analysis determines where changes in texture (class) occur and enables us to distinguish targets from nontargets, manmade objects from natural ones, or tumors from healthy tissue.

Features

We use the three features described earlier, which are derived using the theory of fractal dimension.[7] The fractal dimension D (as distinguished from the normal Euclidean dimension d) can be estimated using Richardson's Power Law[7]

$$M(\varepsilon) = K\,\varepsilon^{(d-D)}, \tag{1}$$

where $M(\varepsilon)$ is the measured property of a fractal at a scale ε, and K is a proportionality constant. This equation and the technique described in Solka et al.[4] allows us to extract the three features. Thus, in a digitized image each pixel can be characterized by a three-dimensional feature vector $\vec{x} = [\,x_1, x_2, x_3]^t$ based on a small neighborhood of the principal pixel. Further, from a single image M, we have available a large sample of observations $X_M = [\vec{x}_1^{\,t},\ldots,\vec{x}_{n_m}^{\,t}]^t$. Using these features, we construct probability density functions (PDFs) for different classes and use these for discrimination.

Probability Density Estimation

The types of problems amenable to these techniques are not those whose PDF can be represented by usual statistical models (e.g., normal distributions). A digitized image can easily

represent a data set of up to 10^7 local observations, and our work indicates this data is not well represented by a normal distribution. We estimate the PDF using a technique such as adaptive mixtures.[8-10] It is a hybrid approach that maintains the best features of the kernel estimation model[11] and the finite mixture model,[12] and does not make strict assumptions about the data distribution. The general mixture density can be given by,

$$\hat{\alpha}(x;\theta,\pi) = \int_{\Omega} \phi(x|\theta)dF_{\pi}(\theta), \qquad (2)$$

where $\hat{\alpha}(x)$ is the estimate for the true PDF $\alpha(x)$ underlying the sample X_M, ϕ is a fixed known function, and F is the mixing distribution.

Segmentation Boundaries

As described by Priebe et al.,[13] we can incorporate segmentation boundaries into the calculation of the fractal dimension features and hence into the PDF. Incorporation of segmentation boundaries provides for significantly more discriminatory information in the texture features and the associated PDFs. This reference further describes the two texture patches from Brodatz[14] shown in Figure 4. Although this may seem to be a trivial case, it is illustrative of the technique. The three regions in Figure 4 (numbered 1 through 3 from the left) show a pure texture (D17 from Brodatz) in region 1 and a pure texture (D24 from Brodatz) in region 3. Region 2 straddles the boundary between the two textures. Figure 5 shows the results of a PDF calculation (single feature) of the Figure 4 regions; α_1 and α_3 are the PDFs of regions 1 and 3, respectively. The two plots of α_2 (region 2) show the effect of incorporating or not incorporating the boundary. Clearly,

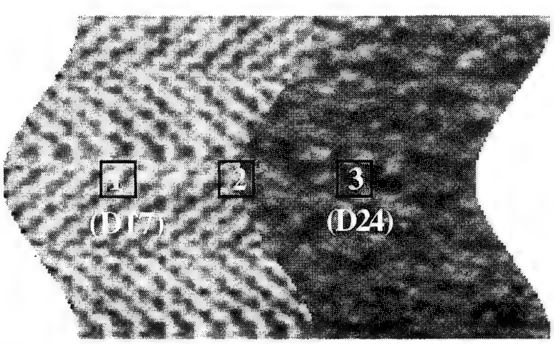


Figure 4. *Two adjacent texture patches and three regions (numbered 1 through 3 from the left).*

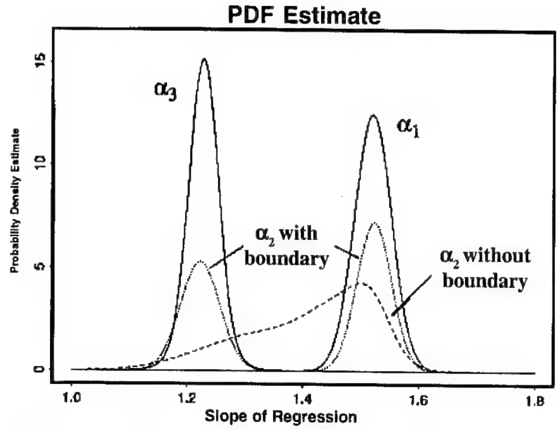


Figure 5. *Single feature PDFs for the three regions from Figure 4.*

incorporating the boundary gives a truer picture of the PDF of the region.

Computational Complexity Reduction

For each observation, the extracted fractal features are represented by $\vec{x} = [x_1,x_2,x_3]^t$. While it is true that more information is often contained in higher dimensional feature space, the computational complexity increases dramatically with any increase in the dimensions.[15] To reduce this complexity and simplify the computations, we use the Fisher Linear Discriminant (FLD).[16] The FLD projects the three dimensions to the one dimension that is, in some sense, best for discrimination. The method and results have been described in Priebe et al.[17] As shown in Reference 17, using all three features and the FLD yields better correlation with class than any single feature alone.

Discriminant Analysis

The PDF characteristics are used to discriminate among the classes[5,18] by a relatively straightforward application of Bayes' rule.[16] Here, we consider

$$X_M = \bigcup_{a \in A_M} X_{M_a}, \qquad (3)$$

where A_M is a set of one or more classes. That is, observations from each image may be drawn from more than one class. In the simplest case, $A_M = \{1,2\}$. Hence, with estimates $\hat{\alpha}_1$ and $\hat{\alpha}_2$ for two classes based on observations X_{M1} and X_{M2} (from image M), the likelihood ratio test statistics, $LR(\zeta) = \hat{\alpha}_1(\zeta) / \hat{\alpha}_2(\zeta)$, are used to indicate the proper classification for the observation, ζ, drawn

from another image. Generalization issues of using estimates from observations of one image for discriminating classes in another image need to be addressed.[19] At a minimum, to discriminate classes in image k (classify the observations in X^k_M), a large number of training observations from images X^i_M ($i=1,\dots,p; i\neq k$) will need to be used to build the estimates $\hat{\alpha}_1$ and $\hat{\alpha}_2$ for the two classes.

Change Point Analysis

Spatial Change Points. With the assumption that an image consists of observations from more than one class, another approach is to investigate the homogeneity of the texture. Considering whether or not the probabilistic structure of an image is uniform throughout may be construed as a spatial change point detection problem.[20] The hypothesis is that there is a region in an image whose probabilistic structure differs from the norm. The investigation of this hypothesis begins by considering small sample regions, $Y_{M_i} \subset X_M, i = 1,\dots,M$. These small sample regions may or may not intersect. Each small sample yields a PDF estimate. From these, we can form a distance function

$$f(\hat{\alpha}_{M_i},\hat{\alpha}_{M_j}) = KL(\hat{\alpha}_{M_i},\hat{\alpha}_{M_j}) = \hat{\alpha}_{M_i} \cdot \log(\hat{\alpha}_{M_i} / \hat{\alpha}_{M_j}) \quad (4)$$

and the statistic

$$T = \sup_{i,j} f(\hat{\alpha}_{M_i},\hat{\alpha}_{M_j}) \quad (5)$$

The integral is the Kullback-Leibler (KL) information between the two distributions and can be used to indicate nonhomogeneity.[19] This is done by estimating the probability density of the KL statistic and using T to distinguish between the homogeneous or nonhomogeneous class. T greater than some τ indicates nonhomogeneity, and estimating the distribution of the T statistic allows a computation of an empirical p-value. This procedure fits into the spatial change point detection framework when each Y_{M_i} is considered to be a spatially connected region. An appropriate value of τ is determined through training; that is, we wish to determine the relationship between T values and the likelihood that an observation deviation indicates a nonhomogeneity.

Spatio-Temporal Change Points. This technique is also useful for detecting changes over time. We can consider images of the same scene or object produced at different times. The characteristics of the regions of the images are modeled by PDFs. A nonhomogeneity in a like region of sequential images indicates a spatio-temporal change point.

Proposed CAD System

Figure 6 shows a proposed system[21] incorporating the items discussed above. This flowchart represents a very high-level schematic.

Experimental Results

Mammographic PDFs

We conducted this study using images provided by the H. Lee Moffitt Cancer Center and Research Institute and the Department of Radiology of the University of South Florida.[17] All tumorous regions were biopsy proven. The mammograms were digitized at ~220 microns/pixel

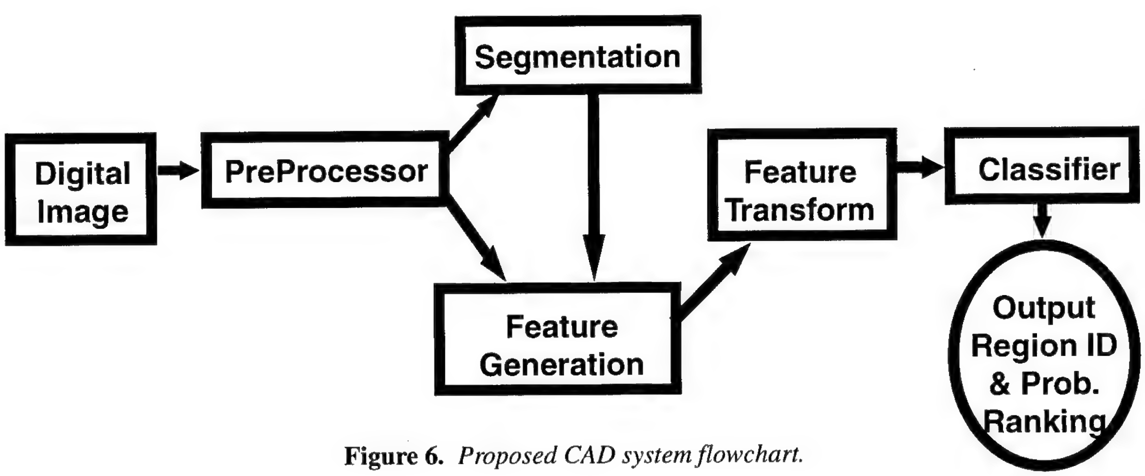


Figure 6. *Proposed CAD system flowchart.*

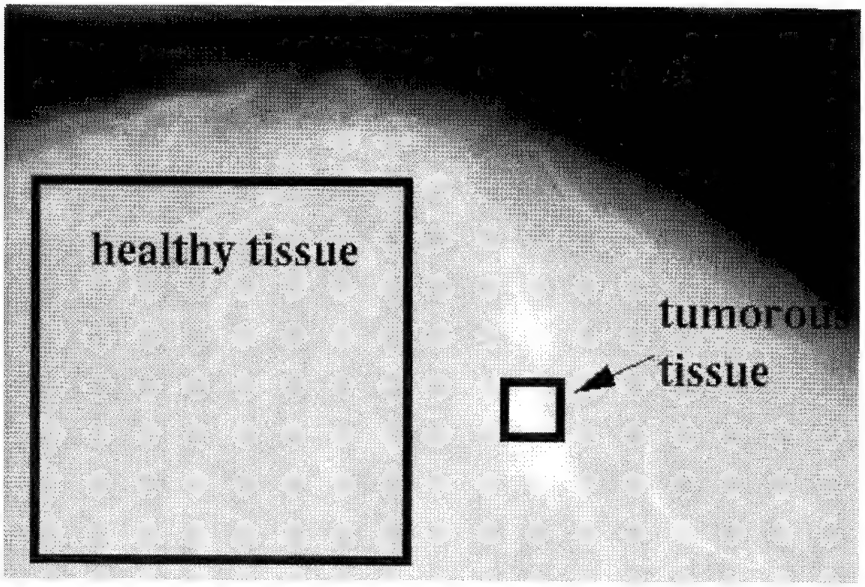


Figure 7. *Regions of interest in mammogram A; this image has been enhanced for presentation.*

and 8 bits/pixel. Figure 7 shows regions of healthy and tumorous (~10-mm malignant stellate mass) tissue from a mammogram A. Training data incorporated 10,000 healthy tissue observations and 500 tumorous tissue observations. A mammogram B (not pictured) containing a ~6-mm malignant stellate mass was used for testing (with 10,000 healthy tissue observations and 300 tumorous tissue observations).

Figure 8 is a plot of the PDFs of the projected data showing the separation of the healthy and tumorous classes for mammogram A. The FLD and transformation from A is applied to B, and the results are shown in Figure 9. The discriminant boundary is clearly evident and appears to be invariant. When the roles of A and B are reversed, the plots exhibit the same behavior but with a different discriminant boundary. Based on this limited study, the results indicate the possibility that once a projection is chosen, the discriminant boundary is invariant from training to testing data. Thus, a discriminant boundary obtained from training images can be successfully applied to new test images.

Wolfe's Patterns

Wolfe distinguished four tissue patterns (labeled as N1, P1, P2, and DY) corresponding to increasing breast tissue density and different

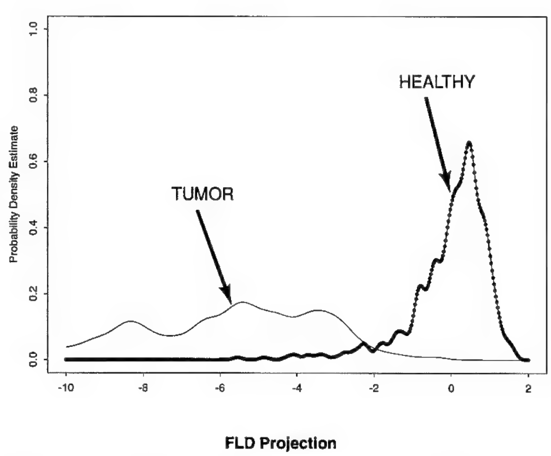


Figure 8. *Fisher Linear Discriminant PDFs for mammogram A.*

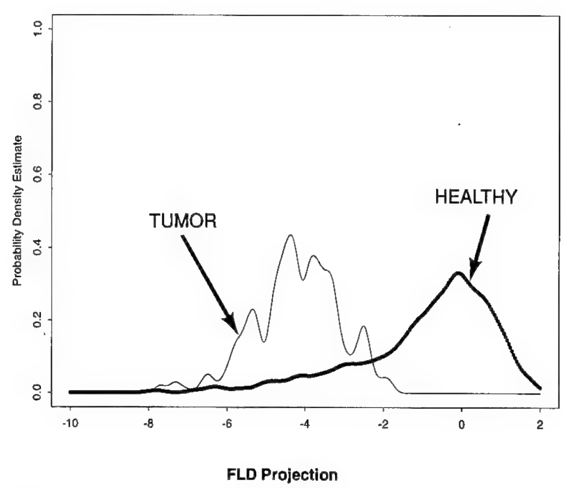


Figure 9. *Fisher Linear Discriminant PDFs for mammogram B using the independent projection.*

morphology.[22] To determine the applicability of this technique to the discrimination of Wolfe patterns, we analyzed an additional eight mammograms from the set provided above. We used two patterns for training data and two others for testing data. Figures 10 through 12 show the PDFs of the patterns indicated. The combinations shown were chosen simply for illustrative purposes. In all cases, the ability to discriminate exists, and the discriminant boundaries generalize from training to testing data. If these results can be extended to nonmalignant abnormal tissue, the technique might be useful in distinguishing these types.

Mammograms and Change Point Analysis

The results to be discussed next involve six patients followed for three years; a biopsy-proven

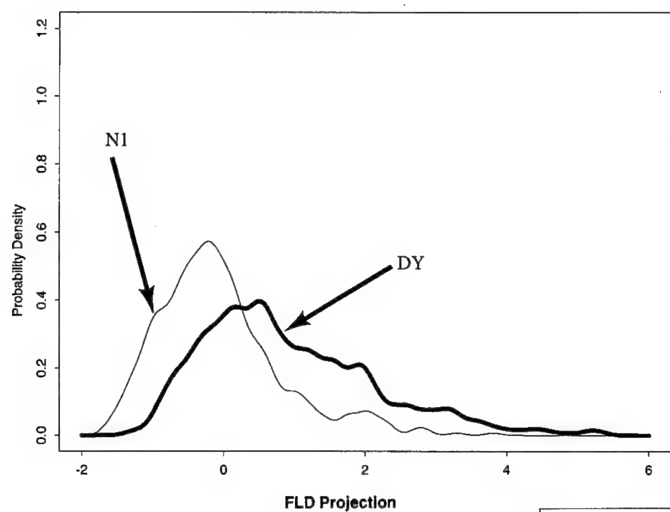


Figure 10. *FLD PDFs for mammogram N1 vs. mammogram DY.*

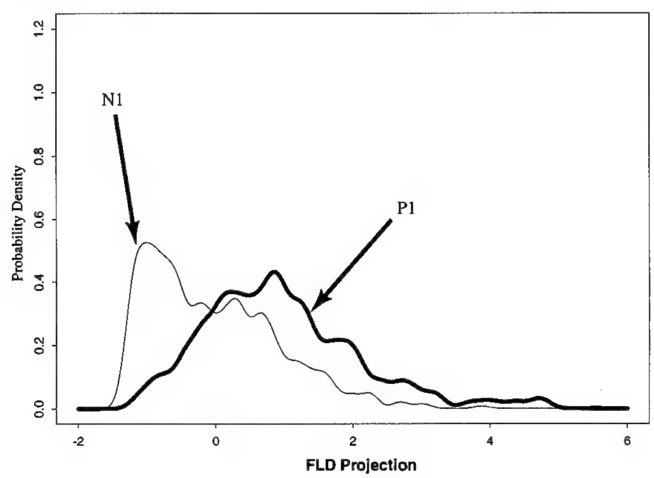


Figure 11. *FLD PDFs for mammogram N1 vs. Mammogram P1.*

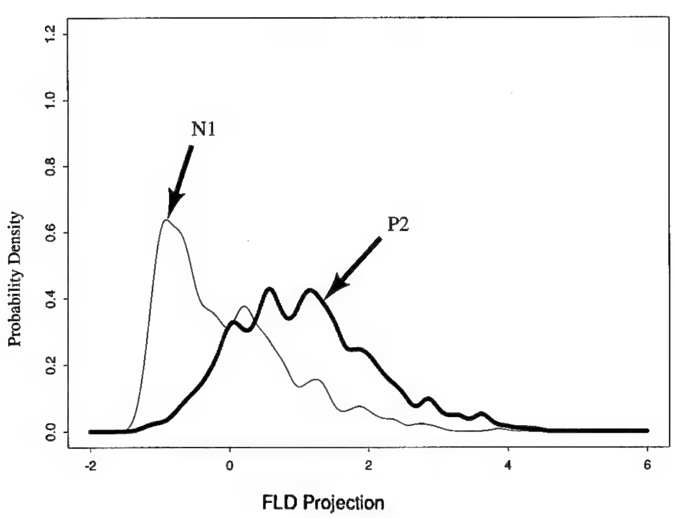


Figure 12. *FLD PDFs for mammogram N1 vs. Mammogram P1.*

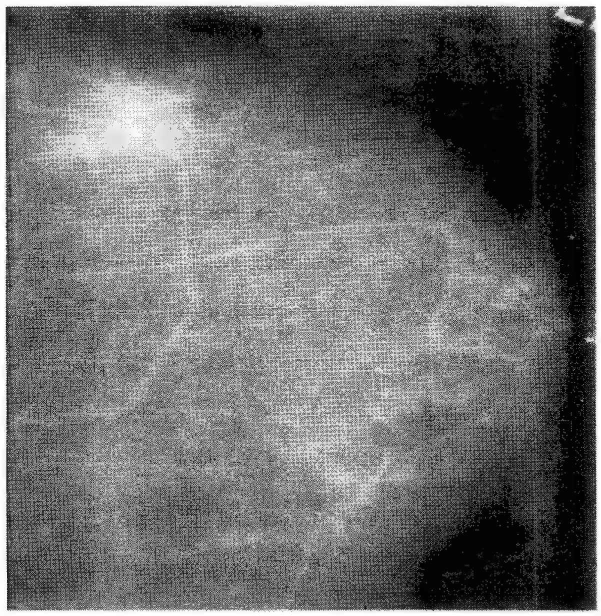
anomaly was detected in the third year in three of the patients. We used at least two views of each breast for each patient for each year, for a total of 81 images. The images were digitized at 600 dpi (~42.3 microns) and 8-bit greyscale. The images were provided by Kaiser-Permanente Research, Portland, Oregon.

We show pictures from only one patient. As will be discussed, we were able to detect an anomaly in the second year. We were not able to do this for the other two cases. However, it may be possible that this technique can result in earlier detection in some cases. We did not detect any false positives in the other three cases.

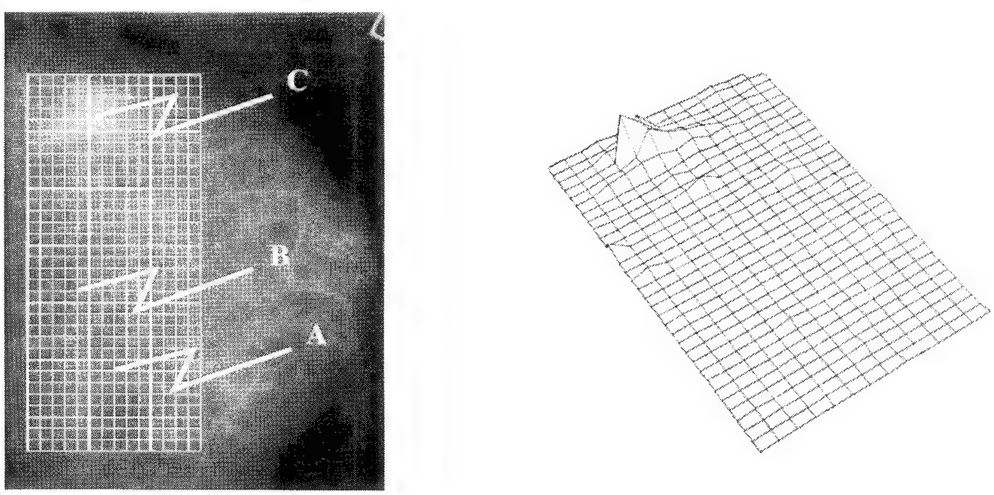
Figure 13 shows:

(a) An image

(b) A grid showing the subregions



(a) A mammogram from year 2. This patient had a tumor detected in the third year of the study.



(b) The mammogram from (a) with the grid overlayed. Observations are drawn from each grid tile. Title A is healthy tissue used as the reference tile. Tile B is another healthy tile. Tile C is in the anomalous region.

(c) Kullback-Leibler surface for the grid shown in (b). The region of large KL values corresponds to the area in which the tumor was detected the following year.

Figure 13. *Mammographic change point analysis.*

(c) The KL surface, $KL(\hat{\alpha}_{ref}, \hat{\alpha}_{i,j})$, for a reference healthy tissue tile against the other tiles i,j.

As mentioned, this image is from the second year. The KL surface appears to be quite homogeneous except at the top, where the tumor was detected in the third year. The KL values are significantly greater here, indicating a region of anomalous tissue.

The PDFs from tiles in the healthy region exhibit similar PDFs, while those in the anomalous region have a shifted mean and a somewhat heavier tail.

If histograms of the KL values are constructed for this patient over the three years, an estimate of a τ value can be made. Using the first year as a baseline healthy set, a $\tau = 2.83$ (maximum KL value) is obtained. For the second year, four detections are obtained; that is, T exceeded τ for four tiles ($T_{max} > 4.5$). For year three, the number of detections was much greater ($T_{max} = 13.67$), which clearly shows the nonhomogeneity detected.

Performance Discussion

Related and Current Work

Breast parenchymal texture characteristics have been studied for their relationship to breast cancer risk as seen in Wolfe[22] and Saftlas and Szklo.[23] Furthermore, texture features have been used for many medical imaging applications,[24-29] including mammography,[30,31] and power law features have proven to be useful in discriminating texture classes in x-ray mammography[17,32-34] as well as in other modalities for breast cancer detection.[35,36] The present work begins from the conjecture that suspicious regions in mammographic images will manifest themselves as distinguishable texture classes, and that these classes will be distinguishable by the fractal-related power law features. While we do not intend this work to be a detailed analysis of the utility of power law features as compared to other texture measures, recent work has indicated that fractal-related measures are a viable texture characterization approach.[37]

The adaptive mixtures approach to estimating the parameters θ and π in $\hat{\alpha}$ (see Equation (2)) allows greater flexibility than standard parametric assumptions and, therefore, holds the promise of superior performance. It also has advantages over conventional nonparametric techniques, such as kernel estimation,[13] in reducing computational complexity. The potential diagnostic value of the subpopulation groupings provided by the resultant mixture estimator may also be useful. As stated above, the utilization of these probability density estimates for the low-level, texture-based information is investigated using discriminant analysis and change point analysis.

The performance of the combination of fractal dimension features using segmentation boundaries and PDFs was analyzed in Priebe et al.[13] The mammogram shown in Figure 14 has a boxed region containing a tumorous region (biopsy-verified) with the radiologist's boundary drawn within it. The tumorous region (region 1) is the region within the radiologist's boundary, and the healthy region (region 2) is the area simultaneously within the box and outside the tumorous region.

Figures 15 and 16 show, respectively, PDFs for the two regions when the true boundary has been incorporated into the calculation of the features[15] and when no boundary is used.[16] We clearly see that the presence of the boundary in the feature extraction is vital to the

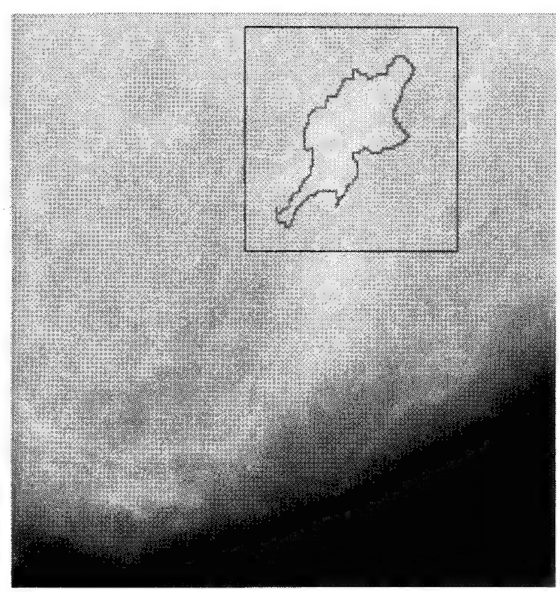


Figure 14. *Mammogram with radiologist's boundary of tumorous region overlaid.*

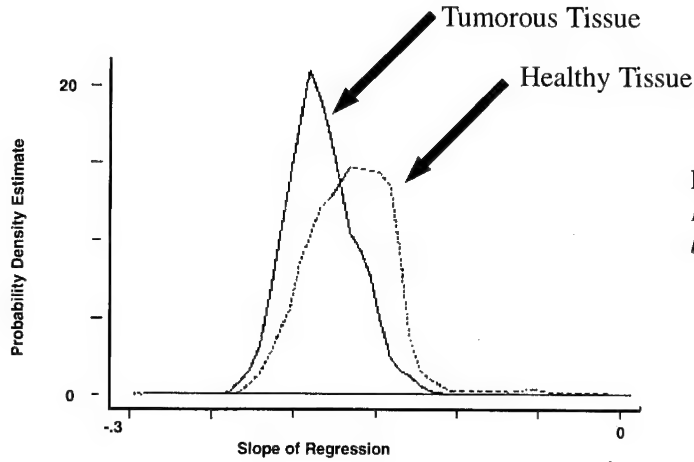


Figure 15. *PDFs for fractal dimension from Figure 14, calculated using the radiologist's boundary.*

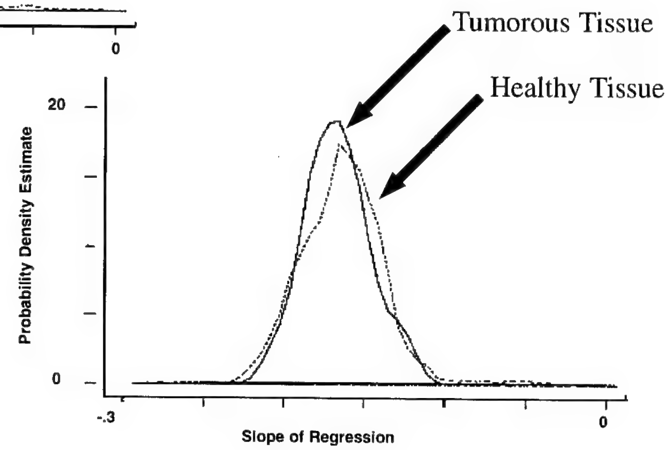


Figure 16. *PDFs for fractal dimension from Figure 14, calculated with no boundary information.*

utility of the features for distinguishing tumorous tissue from healthy tissue.

Unfortunately, obtaining a true boundary like that shown in Figure 14 and used in Figure 15 is costly and time-consuming. Furthermore, the ultimate utility of this procedure for a real application depends on the ability to automatically generate a boundary that will be useful in this context. Figure 17 shows the radiologist's boundary superimposed on a particular wavelet segmentation map. This wavelet map is certainly not perfect. With the boundary, it is not closed, it is not necessarily exactly coincident with the radiologist's boundary, it is continuously valued rather than binary, and there is noise. Nevertheless, it generally marks the edge of the tumorous region. When this boundary is used in the feature extraction, the resultant PDFs are as depicted in Figure 18. We see that the separation of the two classes is maintained to a degree similar to that obtained when the radiologist's boundary was employed. Discriminant analysis could be successfully pursued here, as in Figure 15, while Figure 16 (the no-boundary case) leaves little hope.

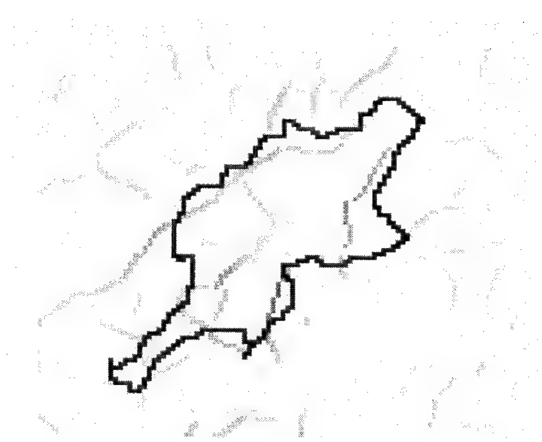


Figure 17. *Incomplete, greyscale wavelet segmentation map with radiologist's boundary overlaid. This continuous valued map is used for Figure 18.*

Future Efforts

The results presented here are preliminary in nature. In fact few studies, if any, have been performed on automated digital mammography processing that are of a large enough scale to draw conclusions about the underlying statistical procedures as opposed to the performance

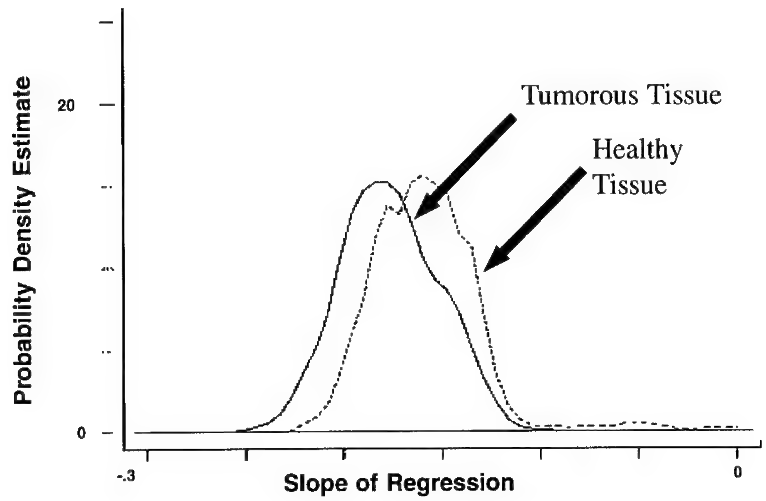


Figure 18. *PDFs for fractal dimension from Figure 14, calculated using the continuous valued wavelet boundary from Figure 17.*

of a system as a whole. Reference 32 presents the study most closely related to the work presented here, but the slant of the paper is significantly different than ours. The studies[38,39] and other related papers by the University of Chicago group investigate radiologist performance using computer-aided technology, but no comparison can be made on the statistical performance of the integral pieces for such a system.

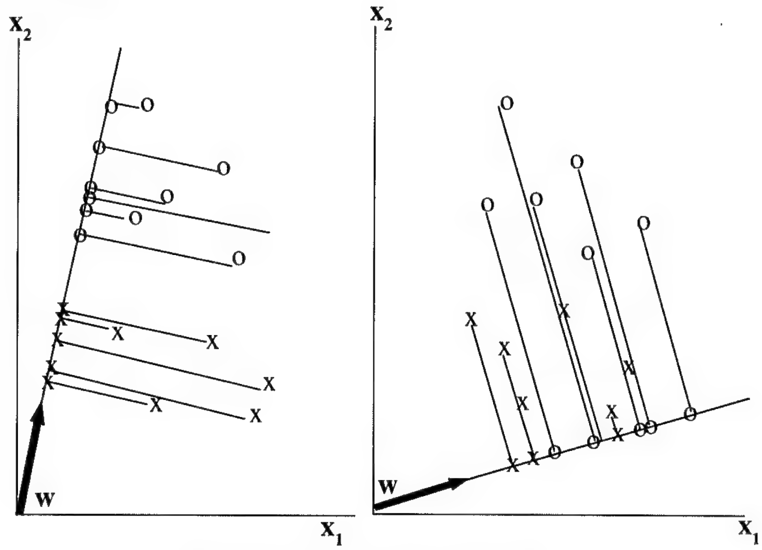
In our opinion, a comprehensive study of the individual processes described here, in the framework of a computer-aided system, is a necessary—albeit complex—next step. The performance of the individual processes cannot be evaluated in a vacuum as there are currently few, if any, useful metrics applicable to the individual processes as opposed to an omnibus computer-aided system.

References

1. Peleg, S.; Naor, J.; Hartely, R.; and Avnir, D., "Multiple Resolution Texture Analysis and Classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 6, 1984, pp. 518-523.
2. Peli, T., "Multiscale Fractal Theory and Object Characterization," *J. Opt. Soc. Am. A*, Vol. 7, 1990, pp. 1101-1112.
3. Pentland, A. P., "Fractal Based Description of Natural Scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 6, 1984, pp. 661-674.
4. Solka, J. L.; Priebe, C. E.; and Rogers, G. W., "An Initial Assessment of Discriminant Surface Complexity for Power Law Feature," *Simulation*, No. 58, 1992, pp. 311-318.
5. Priebe, C. E.; Solka, J. L.; and Rogers, G. W., "Discriminant Analysis in Aerial Images Using Fractal Based Features," *Proc. SPIE*, Vol. 1962, 1993, pp. 196-208.
6. Wegman, E. J., "Computational Statistics: A New Agenda for Statistical Theory and Practice," *J. Wash. Acad. Sci.*, No. 78, 1988, pp. 310-322.
7. Mandelbrot, B., *The Fractal Geometry of Nature*, W. H. Freeman & Co., New York, 1977.
8. Priebe, C. E. and Marchette, D. J., "Adaptive Mixture: Recursive Nonparametric Pattern Recognition," *Pattern Recognition*, No. 24, 1991, pp. 1197-1209.
9. Priebe, C. E. and Marchette, D. J., "Adaptive Mixture Density Estimation," *Pattern Recognition*, No. 26, 1993, pp. 771-785.
10. Priebe, C. E., "Adaptive Mixtures," *Journal of the American Statistical Association*, Vol. 89, 1994, pp. 796-806.
11. Silverman, B. W., *Density Estimation*, Chapman and Hall, New York, NY, 1986.
12. Titterington, D. M.; Smith, A. F. M.; and Makov, U. E., *Statistical Analysis of Finite Mixture Distributions*, John Wiley and Sons, New York, NY, 1985.
13. Priebe, C. E.; Julin, E. G.; Rogers, G. W.; Healy, D. M.; Lu, J.; Solka, J. L.; and Marchette, D. J., "Incorporating Segmentation Boundaries into the Calculation of Fractal Dimension Features," *Proceedings of the 26th Symposium on the Interface*, Research Triangle Park, NC, 1994.
14. Brodatz, P., *Texture: A Photographic Album for Artists and Designers*, Dover, New York, 1966.
15. Scott, D. W., *Multivariate Density Estimation*, John Wiley and Sons, New York, NY, 1992.
16. Duda, R. O. and Hart, P. E., *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York, NY 1973.
17. Priebe, C. E.; Solka, J. L.; Lorey, R. A.; Rogers, G. W.; Poston, W. L.; Kallergi, M.; Qian, W.; Clarke, L. P.; and Clark, R. A., "The Application of Fractal Analysis to Mammographic Tissue Classification," *Cancer Letters*, No. 77, 1994, pp. 183-189.
18. McLachlan, G. J., *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley and Sons, New York, NY, 1992.
19. Priebe, C. E.; Lorey, R. A.; Marchette, D. J.; Solka, J. L.; and Rogers, G. W., "Nonparametric Spatio-Temporal Change Point Analysis for Early Detection in Mammography," *Proceedings of the 2nd Int'l Workshop on Digital Mammography (SIWDM)*, York, UK, 1994.
20. Ripley, B. D., *Statistical Inference for Spatial Processes*, Cambridge University Press, Cambridge, 1988.
21. Rogers, G. W.; Priebe, C. E.; Solka, J. L.; Lorey, R. A.; and Julin, E. G., "A System and Method for Incorporating Segmentation Boundaries into the Calculation of Fractal Dimension for Texture Discrimination," Patent Application, Navy Case No. 75,998, 1994.

Glossary

Fisher Linear Discriminant.



Projection of samples onto a line.

The Fisher Linear Discriminant is a technique designed to reduce a high dimensional problem that is intractable to a low dimensional one that is manageable. Geometrically (as in this example), a vector **w** defines a direction in which the dimensionality is reduced from d dimensions (in this case, two) to the one dimension that is in some sense best. The graph on the left is an example of a projection that results in well-separated samples. The graph on the right is an arbitrary projection that produces a confused mixture of samples. Mathematically, the Fisher Linear Discriminant is defined as that linear function $\mathbf{w}^t\mathbf{x}$ for which

$$J(w) = |m_1 - m_2|/(s_1^2 - s_2^2)$$

is maximum, where the m's are the sample means for the projected points, and the s's are the sample standard deviations. The figures above are from Duda and Hart.[16]

Fractal-Fractal Dimension: Fractals are geometric entities that are said to be self-similar and independent of scale. The idea of self-similarity means that each small portion of a fractal, when magnified, can reproduce a larger portion exactly. Fractals describe natural shapes such as coastlines, clouds, and fern leaves. Fractal dimension need not be an integer as is the case for the more familiar Euclidean dimension. For a fractal dimension between one and two, the fractal fills more space than a simple line but less than a Euclidean area of the plane. The fractal dimension provides a quantitative measure of the curves' wiggliness.

Kullback-Leiber: The Kullback,-Leibler (KL) information is a statistical method used to determine how different two PDFs are. The integral for comparing a probability density f(x) to a probability density g(x) is

$$KL(f,g) = \int f\,(x) \log \frac{f(x)}{g(x)}\, dx.$$

The log of the likelihood ratio, log ((f(x))/(g(x))), is the information in x for discrimination in favor of f against g. Thus, the KL information is the expected discriminatory information and, hence, a measure of the overall discriminatory power of f against g.

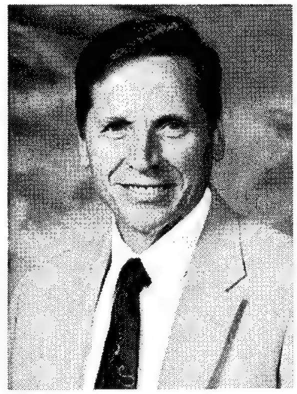
Probability Density Function: The PDF f(x) is defined such that the probability that x is between a and b is given by

$$P(a<x<b) = \int_a^b f\,(x)\, dx.$$

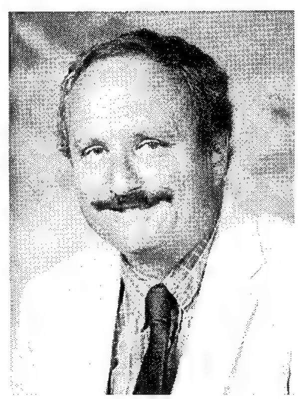
Our goal here is to estimate f(x) for each feature for each class.

22. Wolfe, J. N., "Breast Patterns as an Index of Risk for Developing Breast Cancer," *Am. J. Radiol.*, No. 126, 1976 pp. 1130-1139.
23. Saftlas, A. F. and Szklo, M., "Mammographic Parenchymal Patterns and Breast Cancer Risk," *Epidemiologic Reviews 1987*, Vol. 9, pp. 146-174.
24. Barnsley, M. F.; Massopust, P.; Strickland, H.; and Sloan, A. D., "Fractal Modeling of Biological Structures," *Ann. NY Acad. Sci 1987*, Vol. 504, pp. 179-194.
25. Wu, C-M; Chen, Y-C; and Hsieh, K-S, "Texture Features for Classification of Ultrasonic Liver Images," *IEEE Trans. Medical Imaging 1992*, Vol. 11, No. 2, pp. 141-152.
26. Liang, Z.; Jaszczak, R. J.; and Coleman, R. E., "Parameter Estimation of Finite Mixtures Using the EM Algorithm and Information Criteria with Application to Medical Image Processing," *IEEE Trans. Nuclear Science 1992*, Vol. 39, No. 4, pp. 1126-1133.
27. Cargill, E. B.; Barrett, H. H.; Fiete, R. D.; Ker, M.; Patton, D. D.; and Seeley, G. W., "Fractal Physiology and Nuclear Medicine Scans," *Proc. SPIE 1988*, Vol. 914, pp. 355-361.
28. Cargill, E. B.; Donoho, K.; Kolodny, G.; Parker, J. A.; and Zimmerman, R. E., "Analysis of Lung Scans Using Fractals," *Proc. SPIE 1989*, Vol. 1092, pp. 2-9.
29. Dellepiane, S.; Serpico, S. B.; Vernazza, G.; and Viviani, R., "Fractal-Based Image Analysis in Radiological Applications," *Proc. SPIE 1987*, Vol. 845, pp. 396-403.
30. Kimme-Smith, C.; Frankl, G.; Wassel, G. N.; and Sklansky, J., "Toward Reliable Measurement of Breast Parenchymal Pattern," *Proc. Conf. on Comput. Appl. in Radiol. & Comput./Aided Anal. of Radiol. Images 1979*, pp. 118-121.
31. Miller, P. and Astley, S., "Classification of Breast Tissue by Texture Analysis," *Image and Vision Computing 1992*, Vol. 10, No. 5, pp. 277-282.
32. Caldwell, C. B.; Stapleton, S. J.; Holdsworth, D. W.; Jong, R. A.; Weiser, W. J.; Cooke, G.; and Yaffe, M. J., "Characteristics of Mammographic Parenchymal Pattern by Fractal Dimension," *Phys. Med. Biol. 1990*, Vol. 35, No. 2, pp. 235-247.
33. Magnin, I. E.; Cluzeau, F.; Odet, C. L.; and Bremond, A., "Mammographic Texture Analysis: An Evaluation of Risk for Developing Breast Cancer," *Optical Engineering 1986*, Vol. 25, No. 6, pp. 780-784.
34. Liu, X.; Zhang, G.; and Fox, M. D., "Fractal Description and Classification of Breast Tumors," *Proc. Annual Conf. on Eng. in Med. and Biol. 1991*, pp. 112-113.
35. Goldberg, V.; Manduca, A.; Ewert, D. L.; Gisvold, J. J.; and Greenleaf, J. F., "Improvements in Specificity of Ultrasonography for Diagnosis of Breast Tumors by Means of Artificial Intelligence," *Medical Physics 1992*, Vol. 19, No. 6, pp. 1475-1481.
36. Bronskill, M. J.; Yaffe, M. J.; Boyd, N. F.; Poon, C.; and Byng, J., "Correlation of Quantitative MRI Parameters with Mammographic Pattern: Risk of Breast Cancer," *Proc. Annual Conf. on Eng. in Med. and Biol. 1991*, pp. 108-109.
37. Ohanian, P. P. and Dubes, R. C., "Performance Evaluation for Four Classes of Textural Features," *Pattern Recognition 1992*, Vol. 25, No. 8, pp. 819-833.
38. Astley, S. M. and Taylor, C. J., "Combining Cues for Mammographic Abnormalities," *Proc. BMVC90 1990*, pp. 253-258.
39. Chan, H. P.; Doi, K.; Vyborny, C. J.; Schmidt, R. A.; Metz, C. E.; Lam, K. L.; Ogura, T.; Wu, Y. Z.; and Macmahon, H., "Improvement in Radiologist's Detection of Clustered Microcalcifications on Mammograms. The Potential of Computer-Diagnosis," *Investigative Radiology*, Vol. 25, No. 10, 1990, pp. 1102-1110.

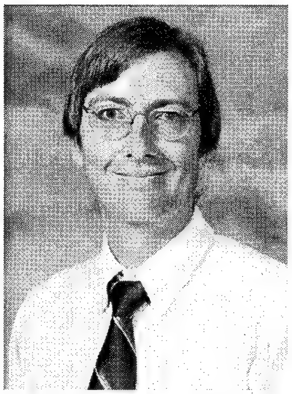
The Authors



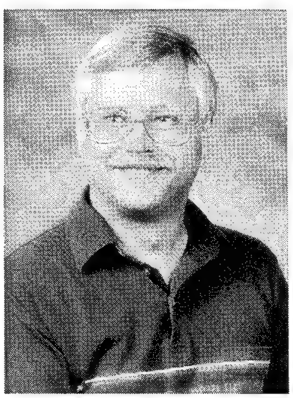
RICHARD A. LOREY earned a B.S. degree in physics in 1962 from the University of Pittsburgh and an M.S. and a Ph.D. in physics in 1967 and 1969, respectively, from the Georgia Institute of Technology. He has been at NSWCDD since 1969. At Dahlgren he has worked in Weapons Safety Research, Range Instrumentation Development, and as branch head of Electronic Warfare-Countermeasures, SLBM Targeting, SLBM Fire Control Development, and Space and Geodesy. He has received the Department of the Navy Meritorious Civilian Service Award. Currently, he is head of the Advanced Computation Technology Group in the Systems Research and Technology Department. His research interests include computational statistics and pattern recognition applied to both military use and medical imagery.



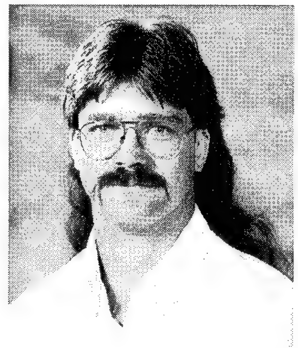
JEFFREY L. SOLKA earned a B.S. degree in mathematics and chemistry from James Madison University in 1978, an M.S. in mathematics from James Madison University in 1981, an M.S. in physics from Virginia Polytechnic Institute and State University in 1989, and a Ph.D. in computational sciences and informatics (computational statistics) at George Mason University in 1995. Since 1984, Mr. Solka has been working in nonparametric estimation and statistical pattern recognition for NSWCDD. He has published over 50 journal, conference, and technical papers; has won numerous awards ; holds 2 patents; and has submitted 2 other applications/invention disclosures.



GEORGE W. ROGERS received a B.S. in physics (Magna Cum Laude) from Georgia Southern University in 1977 and a Ph.D. in theoretical physics from the University of South Carolina in 1984. He came to NSWCDD in 1985 and worked first in satellite orbit computation and more recently in the areas of pattern recognition and computational statistics, including neural networks. He is also an adjunct professor of physics for Virginia Polytechnic Institute and a corresponding professor at the Center for Computational Statistics at George Mason University. He holds patents on neural networks, artificial vision, and computational statistics. He has published over 50 technical papers in computational statistics, pattern recognition, neural networks, and theoretical physics. He currently is the principal investigator for an ONR-sponsored effort in mid-level machine vision.



DAVID J. MARCHETTE earned a B.A. degree in mathematics in 1980 and an M.A. in mathematics in 1982, from the University of California at San Diego. From 1985 to 1994 he worked at the Naval Ocean Systems Center, San Diego, CA, researching pattern recognition and computational statistics. This work has been applied to the areas of Hull-to-Emitter correlation, aerial image analysis, and digital mammography. In 1994 he came to NSWCDD where he has continued his research. He is a co-inventor of the adaptive mixtures algorithm, which is used extensively in computational statistics pattern recognition. He is currently in the Ph.D. program in Computational Sciences at George Mason University. He has authored or co-authored over 50 journal, conference, and technical papers and is a member of IMS, ASA, and AAAS.



CAREY E. PRIEBE earned a B.S. degree in mathematics from Purdue University in 1984, an M.S. in computer science from Diego State University in 1988, and a Ph.D. in information technology (computational statistics) from George Mason University in 1993. He also completed two years of undergraduate work at the United States Military Academy at West Point and graduate work in Mathematics at the University of California at San Diego. Since 1985, Dr. Priebe has been working with nonparametric estimation and statistical pattern recognition at the Naval Ocean Systems Center, San Diego, CA; the Naval Surface Warfare Center Dahlgren Division, Dahlgren, VA; and currently as Assistant Professor in the Department of Mathematical Sciences at The Johns Hopkins University, Baltimore, MD. He is a coinventor of the adaptive mixtures algorithm which is used extensively in computational statistics pattern recognition. He has also authored or coauthored over 70 journal, conference, and technical papers and holds five patents.

Strategic Systems Fire Control

Robert V. Gates

The Naval Surface Warfare Center, Dahlgren Division (NSWCDD) has been an active participant in the Submarine Launched Ballistic Missile (SLBM) Program for nearly 40 years. The initial involvement resulted, in part, from a long history in exterior ballistics and a computing capability that was second to none in the Navy. Over the years, NSWCDD has been in the forefront of advances in trajectory modeling, geodetic systems and models, and computer science. NSWCDD experience in these fields played a key role in the development and targeting of the first SLBM—the POLARIS (A1)—and of every SLBM since. The weapon system requirements for greater range, better accuracy, and increased targeting and operational flexibility have been met, in part, because of NSWCDD advances in computational methods, computer languages and operating systems, and fire control system architecture. Development of the SLBM fire control system of the future will be motivated by different forces than have driven change in the past. Nonetheless, NSWCDD is continuing to use its knowledge and experience in mathematics and computing to anticipate SLBM weapon system needs and to propose innovative fire control and targeting solutions.

Introduction

On 15 November 1960, USS *George Washington* (SSBN 598) departed Charleston, SC, on the first nuclear deterrent patrol. It carried, in addition to 16 POLARIS A1 missiles, some 300,000 target cards prepared by the U.S. Naval Weapons Laboratory (now NSWCDD). Thirty-five years and some 3000 patrols later, fleet ballistic missile submarines (SSBNs) continue to deploy with fire control and targeting products developed by NSWCDD. The Division's expertise in mathematics and computing provided the basis for the initial support of the Special Projects Office (SPO) in 1956. It is still a primary reason that the Division has been able to develop fire control and targeting systems that have allowed full usage of the inherent capability of each of the successive generations of SLBMs (see Figure 1). This article will provide an overview of the technological advances in SLBM fire control and targeting from POLARIS to TRIDENT II and will conclude with a preview of planned and possible changes for the future.

POLARIS

In November 1955, the Secretary of the Navy established the SPO to investigate the unique problems associated with launching an intermediate-range (1500 NM) ballistic missile from a ship. The Army Ballistic Missile Agency was given the responsibility of developing the missile

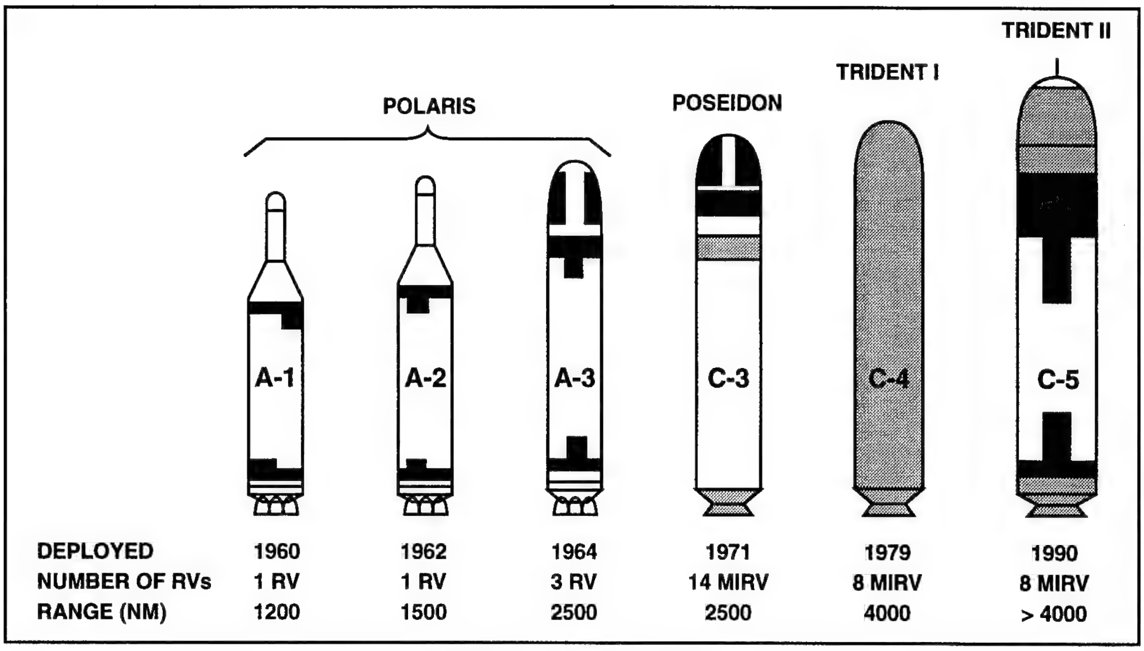


Figure 1. *SLBM evolution.*

and the land-based launching system. By the end of December, an operational requirement was issued, and specific research and technology needs were identified. Throughout much of 1956, the Army and the Navy studied the Army's JUPITER missile for ship and land-based use. NSWCDD's participation began during this period. By the end of the year, the JUPITER concept was discarded, and a Navy concept for a small, solid propellant missile was authorized by the Secretary of Defense. It was decided (by February 1959) that an interim 1200-NM capability (POLARIS A1) would be provided by late 1960 with full 1500-NM capability (A2) in mid-1962. An advanced 2500-NM capability (A3) was required by late 1964.

Dr. Russell Lyddane and Mr. Ralph Niemann visited SSP and presented Dahlgren's capabilities in exterior ballistics and computing. Dahlgren was the Navy's expert in classical exterior ballistics and had produced bombing and range tables since before World War II. Dahlgren's computer resources were also unmatched in the Navy. The Naval Ordnance Research Calculator (NORC), the Navy's foremost digital computer, had been installed in 1955, replacing the Aiken Relay Calculator (MARK II and MARK III). Dahlgren possessed unique expertise in trajectory modeling, having developed what is believed to be the first six-degree-of-freedom trajectory simulation (of a 12.75-in. rocket) in 1950.[1] At about the same time, Dahlgren was supporting the Naval Research Laboratory (NRL) development of the Vanguard satellite through efforts in orbit analysis.[2] This and associated research concerning modeling of the earth's gravity field were aspects of Dahlgren's unique mathematical and computational capabilities, which supported the early POLARIS studies. The earliest of these studies involved the evaluation of guidance presetting methods in support of Q-matrix guidance developed by the Massachusetts Institute of Technology (MIT) Instrumentation Laboratory.

Organizational Evolution–When NSWCDD first began supporting the POLARIS program, we were known as the Naval Proving Ground. By the time the system deployed, we were the Naval Weapons Laboratory. Later, we became the Naval Surface Weapons Center. Similarly, Strategic Systems Programs (SSP) began as Special Projects Office (SPO), became the Strategic Systems Project Office in 1968, the Strategic Systems Program Office in 1984 and, finally, SSP in 1987.

Q Guidance–Q guidance is a form of implicit (i.e., does not require knowledge of missile position) guidance developed by Laning and Battin at the Massachusetts Institute of Technology (MIT).[3] This scheme uses the elements of the Q matrix (which are the partial derivatives of the components of the correlated velocity with respect to the components of the position vector) to compute the required change in the velocity to be gained and, thus, to update an initial estimate of velocity to be gained. When the cross product of the velocity to be gained and its rate of change are nulled, thrust is terminated, leaving the missile on a ballistic trajectory to the target. This concept minimizes in-flight computation and does not require an in-flight gravity model. Both were significant attributes since sufficiently capable and reliable flight computers were beyond the state of technology.

These early studies of the POLARIS missile and its guidance led to Dahlgren being assigned the role that it has filled for every SLBM system since—providing the development of fire control and targeting products. It should be noted that one of SSP's key maxims was that naval laboratories were to be used in the development effort only if their technical competence was not available in private industry.[4] Inherent technical capability and the availability of computing resources were a prerequisite for being assigned a role in POLARIS; however, demonstrating (and continuing to demonstrate) technical competence to SSP was the key factor.

POLARIS Fire Control

In general terms, SLBM fire control:

- Initializes missile guidance with navigation and targeting data
- Aligns and erects the inertial guidance system (i.e., determines the direction of north and vertical)
- Checks the status of other shipboard systems
- Controls the launch sequence

The presetting data for POLARIS were basically the elements of the Q matrix—it was shown that once they are computed from launch and target coordinates, they can be treated as constants for typical POLARIS ranges—and an initial value of velocity to be gained. Computation of these data, which are used to direct the flight of a ballistic missile to the intended target, requires a suitable trajectory model, earth and atmospheric models, and appropriate target information. If this missile is to be fired from a moving platform, these computations will ideally be done immediately before launch using real-time navigation inputs.

In the late 1950s, when Dahlgren was addressing this problem, computer technology did not support this approach. Computers were too large for shipboard installation and too unreliable to be placed in the critical path for launching a weapon. An alternative was to provide direct input of precomputed initial conditions for a large number of possible launch-target point combinations.[5] This approach had two significant shortcomings: it required the submarine to carry a very large amount of data (in the form of punched cards), and computing these data would take an extremely long time on the NORC. Each trajectory calculation on the NORC took 1½ hours of computer time, and it was estimated that 40 years would be required to prepare all of the cards needed for the first patrol.

The solution developed by Dahlgren was a modification of the precomputed data approach. The launch area was divided into 20-NM squares and the target area into 30-NM squares. Presetting data were computed for each of the required pairs and provided to the submarine on punched cards. The data required to interpolate for points within the cells were also provided. Even with this reduction, however, the computational burden on the NORC was still excessive—a large number of trajectory simulations was required. Dahlgren mathematicians solved this problem by running only enough trajectories to develop numerical functions for each of the guidance presettings in terms of launch-point coordinates and target range and bearing. These functions were used to generate the data

transferred to the submarine on target cards. Data were read from the appropriate card (see Figure 2) and entered into the Mark 80 Fire Control System (FCS) using knobs and dials on the input panel. The target card also contained the solution to a test problem, which was used to verify the manual knob settings. This process, as unwieldy as it was, proved to be successful. Dahlgren provided target cards for all operational patrols and for all guided flight tests.

Dahlgren developed or initiated two improvements to the system to address the logistics problems. The first of these was to provide the target cards on microfilm (three cards per frame). A film reader and keypunch were placed on the SSBN so that the crew could produce cards as needed. When the boats were deployed with the A3, an additional upgrade was required, and the Mark 148 POLARIS Target Card Computer System (PTCCS) was developed. The PTCCS was a stand-alone system (not part of fire control) that used Dahlgren-provided programs and data to produce POLARIS target cards. This system (which had 8000 words of memory and averaged 66,000 operations per second) was used until 1982 when the last of the original 10 POLARIS submarines was withdrawn from service.[6]

Mark 84 Fire Control

The Mark 80 FCS was installed on the first ten submarines; it was replaced by the Mark 84 on the 31 Lafayette-class SSBNs. This system, which used the first digital fire control software developed by Dahlgren, became operational in 1963 with the A2 missile. The heart of the system was the Digital Geoballistic Computer (DGBC). It consisted of two Digital Control Computers (DCCs)—a militarized version of a CDC 1604 commercial computer—with access to a common magnetic drum, printer, and punched tape reader. Each DCC had 16,000 words of core memory and averaged some 87,000 operations per second. Dahlgren program and data updates were delivered to the submarine on punched tape and loaded on the magnetic drum.

The FCS performed real-time fire control computations and controlled initialization of the guidance systems for 16 POLARIS missiles. In general terms, the complex POLARIS presetting functions, which were previously solved at Dahlgren to produce target cards, were now solved by the FCS. The results were based on real-time navigation inputs and were periodically updated. The Mark 84 is no longer in service in the U.S. SLBM force. The U.K. signed an agreement with the U.S. in 1963 to purchase the POLARIS A3 system. The Mark 84 FCS- and Dahlgren-produced software are still in service with the U.K. SLBM force.

POSEIDON (C3)

As early as 1962, SSP (and others) began considering a follow-on to A3. The first concepts addressed increased payload at the same range as A3. The larger missile being proposed

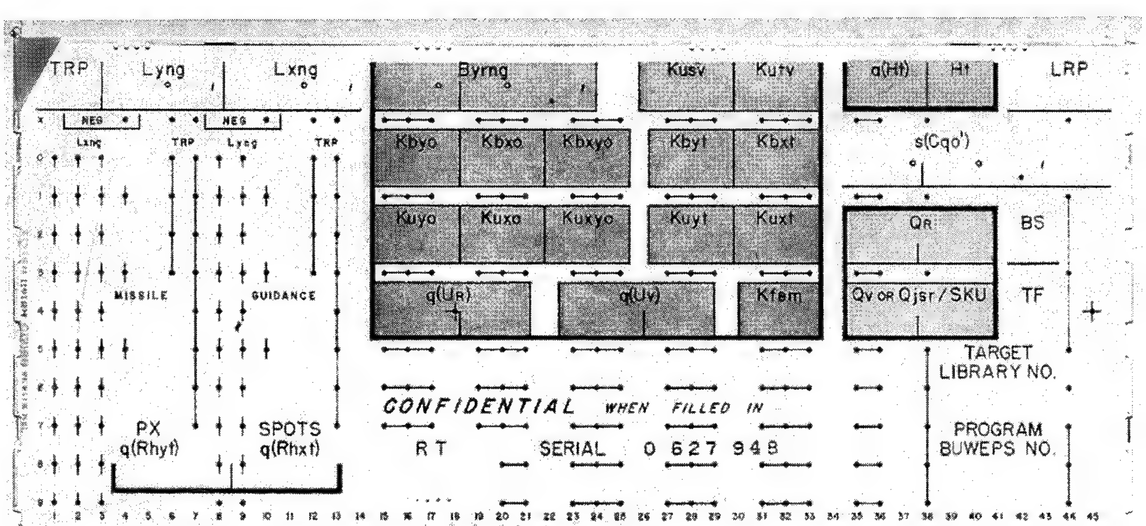


Figure 2. *POLARIS Mark 80 FCS target card.*

was called POLARIS A3A. There was some concern in DoD as to whether or not the A3A would satisfy long-term requirements.[7] During 1963 and early 1964, a variety of reentry systems were considered. In 1964, DoD decided that the next SLBM would be designed to be effective against urban-industrial targets and that it would carry the Mark 12 reentry body (RB) being developed by the Air Force. The resulting missile was larger than the A3A and was referred to as the POLARIS B3. Secretary of Defense Robert S. McNamara recommended development of the B3 to the President in December 1964. In January 1965, President Lyndon B. Johnson announced the development of the next generation SLBM—the POSEIDON C3. The name was apparently changed to emphasize that this was a new system. The POSEIDON C3 became operational in March 1971 when USS *James Madison* (SSBN 627) deployed on an operational patrol.

Studies performed during 1964 indicated that the B3 could carry up to six Mark 12s (compared to one RB on A1 and A2, and three on A3). Several RB deployment schemes were considered. The first choice was to deploy them in a pattern around the target point as was done in A3. The A3 ejection mechanism did not allow a large enough RB separation at the target, and other concepts were proposed. The top candidates were known as *Mailman* and *Blue Angels*. *Mailman* proposed to put a guidance and propulsion system on a post-boost vehicle (or bus) that would carry all of the RBs and release them one at a time to achieve the desired pattern at the target. *Blue Angels* required that each RB have its own guidance and propulsion system. One significant drawback with *Mailman* was that Q-matrix guidance could not be used, because explicit knowledge of missile position was required to properly deploy the individual RBs. *Blue Angels*, on the other hand, would retain Q-matrix guidance. *Mailman* was considered the more elegant solution and was chosen.

Modeling Earth's Gravity Field

Dahlgren became involved with these studies during 1964. Among the first issues were determining the proper guidance algorithm and identifying the associated guidance presettings to be computed in fire control. One implication of explicit guidance is that the in-flight guidance system must use a model of the earth's gravity in its calculations. Since the late 1950s, Dahlgren had been active in orbit determination and, in 1960, used Doppler observations of the Transit 1B satellite to verify the "pear shape" of the earth's gravity field.[8] In the early 1960s, Dahlgren pioneered the development of what was called a "General Geodetic Solution," which provided the simultaneous determination of gravity coefficients, ground tracking station coordinates, and an assortment of sensor and measurement system biases. These preliminary solutions led to the development of the standard Department of Defense (DoD) gravity model—the World Geodetic System 1966 (WGS-66). (Dahlgren has continued to develop this system. Later versions, WGS-72 and WGS-84, have also been DoD standards and were used in later SLBM systems.)

These developments and the POLARIS fire control experience put NSWCDD in a unique position to solve the guidance gravity model problem. The solution proposed by NSWCDD utilized the capabilities of both fire control and in-flight guidance. Guidance used Keplerian equations with an inverse square (or round earth) gravity model for in-flight calculation of position and steering commands. Fire control compensated for the inherent error (due to both the simplified gravity model and guidance's lack of an earth atmosphere model) by calculating offsets to be added to the target vector used in the guidance computations. These offsets (or "Kentucky Windage") are a function of launch point, target point, and the specific trajectory to be flown; and require modeling the guidance computations in a trajectory simulation with higher fidelity gravity and atmosphere models. Ideally, this computation would be done in fire control using real-time navigation inputs. However, this was not possible, and an approximate method was developed for fire control use.

Mark 88 Fire Control

The Mark 88 Mod 0 (and, later, Mod 1) FCS was developed for C3 and replaced the Mark 84 on the 31 *Lafayette*-class SSBNs (see Figure 3). This system closely resembled the Mark 84 but

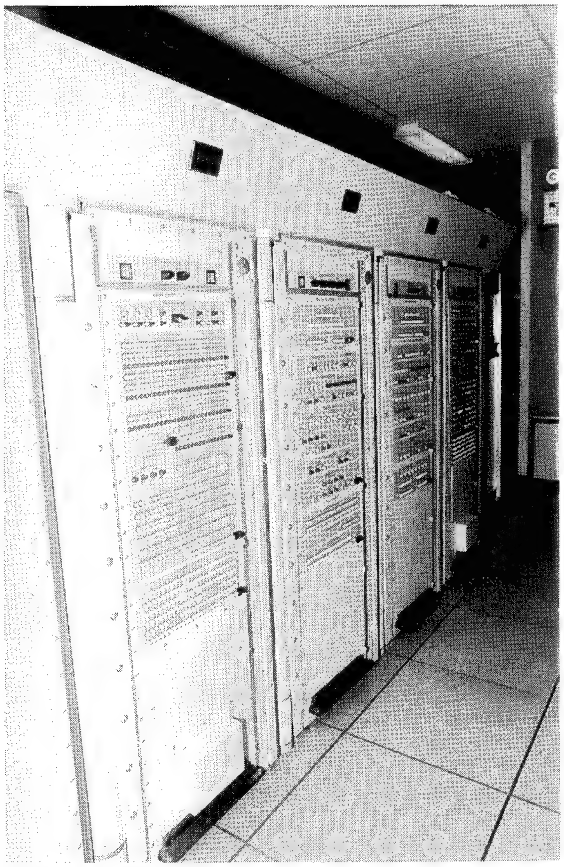


Figure 3. *POSEIDON Mark 88 Fire Control System.*

had several significant additions. The shared magnetic drum was replaced by two magnetic disk file systems, which provided more than an order of magnitude more mass storage, and a keyboard was added to improve the operator interface. The basic computing power, however, was much like that of the Mark 84. Fire control programs and data were sent from Dahlgren to the submarine on magnetic disk packs. Targeting data could be changed onboard the SSBN by entering data (by keyboard or punched tape) to a reserved area on the disk pack.

The fire control presetting calculations were estimated to be an order of magnitude more complex than those for POLARIS, and 20 times the number of guidance presettings were computed. This was the result of the more complicated guidance scheme used in C3 and the fact that there were now multiple independently targetable reentry vehicles (MIRVs) to be considered. Fire control computed the booster presettings and the offsets for each RB target, and checked the achievability (i.e., sufficiency of bus energy to release all RBs with required velocity). In previous systems, all flights were on minimum energy trajectories. POSEIDON fire control provided the capability to select a time-of-flight option. As before, fire control allowed real-time control of the system by providing computer control of the guidance erection and alignment process, by providing the interface between navigation and the missile, and by checking the status of the shipboard systems required to launch the missile. Dahlgren's growing expertise in computer science was the key to finding a way to solve the more complex POSEIDON fire control problem in what was basically a POLARIS computer. This solution included developing efficient algorithms and applying innovative methods in computer science. Dahlgren developed a unique computer operating system (the POSEIDON SUPERVISOR) that controlled relocatable programs (i.e., managed memory) in order to simultaneously prepare all 16 missiles to be launched. In the end, there was actually an increase in overall FCS performance.

At the same time, the Joint Strategic Targeting Planning Staff (JSTPS) was wrestling with the problem of targeting a MIRV system. Previously, targeting was primarily a matter of assigning a target to the one warhead on the missile and making sure that the target was within the range of the missile. The MIRV problem was much more complex. Each warhead on the missile had to be assigned a target, and a "footprint" had to be developed. Maximizing the separation of the

In a MIRV system, each warhead is assigned to an "aimpoint" (or target), and all aimpoints assigned to warheads on the same missile are collected in "footprints." Footprints are an ordered set of aimpoints and are constrained in geographic extent by the capabilities of the equipment section and the trajectory shape. Multiple footprints are collected in "target packages." All footprints in a target package will be struck by missiles from the same SSBN.

JSTPS –The JSTPS was established by President Eisenhower and Secretary of Defense Thomas S. Gates in August 1960. It was located at Offutt Air Force Base (AFB) in Omaha, Nebraska. The Director was the Commander in Chief of the Strategic Air Command, and the Vice-Director was always a Navy flag officer. This function is now performed by the U.S. Strategic Command.

individual ground targets required determining the optimum sequence in which they should be struck (and thus, how they should be assigned to RBs on the bus, because RBs are released in a specified order to preserve bus stability). This was not a job that could be accomplished with the simple tools available to JSTPS. Suitable computer models were developed at NSWCDD and supplied to JSTPS. Dahlgren has provided such models and guidance for all SLBM systems ever since.

Quality Assurance and Configuration Management

Dahlgren's other major contribution to POSEIDON was the development of a quality assurance and configuration management system for fire control and targeting products. While this was done on earlier systems, the stringent testing and configuration control that characterizes SLBM has its origins in POSEIDON. The process has evolved with each of the successive systems and, at SSP's direction, has been applied by other SLBM contractors. It was a model for similar efforts (such as TOMAHAWK) at Dahlgren.

TRIDENT I (C4)

While POSEIDON development was underway, consideration of the next generation SLBM system began. In late 1966, a study called "STRAT-X" examined alternatives to counter a Soviet antiballistic missile (ABM) threat. The Navy concept that emerged from this study was known as the undersea long-range missile system (ULMS). It was a larger missile than POSEIDON or POLARIS and would require the development of a new submarine. By 1971, two specific alternatives had emerged: ULMS and a new submarine, or an extended-range POSEIDON (called EXPO) that could be carried on Lafayette-class SSBNs. It appeared (based largely on submarine construction schedules) that ULMS could not be deployed until the early 1980s (possibly as late as 1983). EXPO, on the other hand, could be fielded in late 1977. Dahlgren supported SSP and the Chief of Naval Operations' (CNO) staff in defining the basic requirements for this new SLBM system.

The Secretary of Defense announced his ULMS decision on 14 September 1971. ULMS I would be a 4000-NM missile that was compatible with the POSEIDON submarines. ULMS II would be a longer range missile to be deployed in a new submarine. ULMS I would be deployed in 1977; no specific deployment date was set for ULMS II. ULMS was renamed TRIDENT in May 1972. The TRIDENT I C4 became operational on USS *Francis Scott Key* (SSBN 657) in October 1979 and on USS *Ohio* (SSBN 726), the first large submarine, in October 1982.

SSP resisted (as they had on previous systems) setting accuracy objectives for C4. Instead, the goal was to increase missile range to 4000 NM, while maintaining C3 accuracy. The longer range was needed to increase sea room for the submarine in order to counter the Soviet antisubmarine warfare (ASW) threat. Accomplishing this required that the system be more accurate, as target miss tends to increase with range. One of the key changes to the system was the addition of a stellar sensor to the guidance system; another was higher fidelity fire control compensation for gravity effects. Dahlgren had a hand in the development and implementation of both.

TRIDENT I takes a star sighting before release of the reentry bodies. A preselected star is located, and two error coordinates are measured. These coordinates, which represent angular rotations about two of the three axes of the guidance coordinate frame, are combined with a precomputed weighting (W) matrix (based on statistical estimates of weapon system errors) to estimate guidance position,

velocity, and orientation errors. These estimates are then used to update the guidance computer. Some of the early work on stellar guidance began during the development of C3. Dahlgren contributed to the analysis of its accuracy potential and, in particular, of the operational implementation issues.

Dahlgren's contribution to stellar guidance took two specific forms in addition to the more general concept analysis. These were the development of the fire control computations required to select the optimum star for accuracy and compute the W matrix for a given launch point and target point combination, and the development of the operational star catalog. Both star selection and W-matrix computation are based on knowledge of star location, weapon system error sources and modeling, and trajectory conditions (including launch and target coordinates). Since star position is a function of time of day, launch point, and trajectory conditions—all of which change as the submarine moves—these computations must be performed in fire control near the time of launch (or performed in such a way that they are relatively insensitive to changes in time or position). Further, they had to be defined so that they maintained the readiness time and launch rate of POSEIDON. These computations were all developed by Dahlgren and met all timing and accuracy requirements.

The operational star catalog had to meet certain requirements—included stars had to:

- Exceed a minimum brightness
- Have a relatively constant brightness
- Have a minimum separation from other stars
- Have a predictable position

It turned out that there was no star catalog that met all of these requirements. Dahlgren obtained several of the standard catalogs and analyzed and compared them. They did not always contain the same stars or, if they did, there was not always agreement on position data, brightness, or the coefficients used to update star position from the reference epoch to the current time. Dahlgren resolved many of the discrepancies and produced the "Dahlgren General Catalog." This, in turn, was used to select the subset of stars that constitute the C4 operational catalog.

TRIDENT I fire control is also distinguished by the fact that it uses an onboard trajectory model to compute guidance presettings. Previous fire control computations compensated for the oblate gravity terms in the model of the earth's gravity. Tesseral gravity effects were compensated for as part of the target offset functions derived at Dahlgren. Maintaining C3 accuracy at the longer C4 ranges required more accurate compensation at the tesseral gravity level. Further, this compensation and other presetting computations led to the need for a trajectory model in fire control in place of the evaluation of functions developed at Dahlgren. A basic requirement, noted above, was that readiness time and firing rate could not be affected. The development of a fast (and sufficiently accurate) trajectory model that executed on the TRIDENT I fire control computer was a major accomplishment.

Mark 98 Fire Control

The Mark 88 Mod 2 FCS was developed for use with C4 on the backfitted POSEIDON submarines. The Mark 98 Mod 0 FCS was used with C4 on the larger USS *Ohio* (SSBN 726) class TRIDENT submarines. Dahlgren was intimately involved in determining the design characteristics and architecture for both of these FCSs. The design used the Mark 88 technology as a base and added some significant improvements. These included replacing the DCC with a more capable computer known as the TRIDENT DCC (TDCC) and adding

Earth's Gravity Field –The earth's gravity field can be represented as a series of spherical harmonics. The first term is the inverse square (or, round earth model). Adding the second (the "J2") term causes the earth to be represented as an oblate spheroid. At NSWCDD (and in SLBM), the next set of terms (up to degree and order nine) are often referred to as the "tesseral" gravity terms. The remaining very localized effect is generically called "high-frequency" gravity.

magnetic tape cartridges (MTCs) to supplement the magnetic disk packs. The disk packs were no longer adequate to store all of the required data, and the capability was added to read targeting and geophysical data from the MTCs. In addition, fire control test data and some guidance data are provided on MTCs.

Perhaps more significant overall were the fire control software changes implemented for TRIDENT I. Based on the expected complexity of the C4 fire control software, NSWCDD recommended to SSP that some fundamental changes were required. These included a new real-time operating system and the use, for the first time, of a higher level programming language. The new operating system was developed completely by NSWCDD. It allowed partitioning of the software and met all of the real-time support requirements. (Much of this was in support of the erection and alignment of the guidance system. This was digitally (software) controlled in C4 rather than analog, as in previous systems.) NSWCDD also developed a nonintrusive measurement and debug system that extracted data from the FCS in real time. This system, the Verification and Evaluation System for TRIDENT (VEST), was the model for the same capability in the UYK-43, a standard shipboard computer in the surface navy. Dahlgren developed the TRIDENT Higher Level Language (THLL) used in C4 fire control as well as the associated compilers, linkers, loaders, and other support software. Fire control software for previous systems was written in machine language. It was estimated that it took two to three years to become proficient at this. Thus, another benefit was the relative quickness with which new employees could contribute to the development. This was aided further by the addition of structured software techniques to the software development process at Dahlgren. NSWCDD was in the forefront of developments in structured programming and quality assurance during this period.

TRIDENT II (D5)

The CNO, ADM Elmo R. Zumwalt (in 1972) and the Secretary of Defense, James Schlesinger (in 1973), asked for information on possible accuracy improvements to the SLBM system. Schlesinger, in particular, felt that the nation's security needs could be better satisfied with a more accurate system. SSP was reluctant to commit to a more stringent accuracy requirement, because they lacked the ability to measure error contributions and the understanding required to extrapolate results to other than the test conditions. The Improved Accuracy Program (IAP) was the result of discussions between Schlesinger and the Director of SSP, RADM Levering Smith. Spanning from 1974 to 1982, this program had several objectives:

- Gaining an understanding of SLBM error sources
- Assessing the accuracy potential of improved components and concepts
- Starting advanced development of promising components and concepts

A major thrust was developing new instrumentation methods so that the needed error source data could be gathered.

Dahlgren participated fully in the IAP program. New concepts such as the "Inverted Global Positioning System (GPS)" (a system whereby the then-incomplete GPS constellation could be augmented by ground stations), GPS-aided guidance, and terminal sensors on reentry bodies, represent some of the concepts investigated by Dahlgren to assess accuracy potential or to identify operational issues. Improved fire control methods, such as high-frequency gravity modeling and compensation, were developed. Similarly, NSWCDD contributed to the investigation of improved stellar guidance concepts. Flight tests were supported, either by mission planning and postflight analysis or, in the case of the SATRACK system, by producing precise GPS ephemerides for postflight estimation of errors. One of the major lessons of IAP (and one that Dahlgren contributed to learning) was that a total-system approach was required to develop a very accurate system. It was no longer sufficient to optimize accuracy at a subsystem level. A systems-engineering approach based on the specification of system and subsystem error budgets verified by precise

measurements and computer simulation (first developed for TRIDENT I) was expanded and used for TRIDENT II.

In 1977, Congress authorized funding for initial TRIDENT II (D5) studies. There were a number of issues to be resolved including missile configuration (i.e., payload weight and type) and accuracy requirements. SSP's initial desire was to carry the C4 reentry body (the Mark 4) in greater numbers or to a greater range. There were external pressures to develop a more accurate missile. Finally, in October 1981, an advanced development program (for a late 1989 Initial Operational Capability (IOC)) was authorized. The system was to carry a new higher yield Mark 5 reentry body (while maintaining the capability to carry the Mark 4) and to be highly accurate. IOC for the TRIDENT II D5 was achieved in March 1990.

A number of system changes were needed to achieve the required accuracy. These included modifications to the guidance system (including a new inertial measurement unit (IMU) and a new stellar sensor), a new navigation system (Electrostatically Supported Gyro Navigation (ESGN) instead of ship's inertial navigation system (SINS)) as well as other modifications to the navigation system (such as a Navigation Sonar System to measure velocity), and a new equipment section (bus) and RB release mechanism. Fire control computations were also changed. A more accurate compensation of gravity (including high-frequency gravity) was required. Compensation of reentry wind and density effects and the fire control computations, in general, were made more accurate. Changes were also required to support the new stellar sensor. These included the development by Dahlgren of a new weighting matrix and update scheme and a new operational star catalog. A key, as highlighted previously, was that the fire control software had to be designed to pull the entire weapon system together to achieve overall goals.

Dahlgren innovation in gravity modeling is particularly noteworthy. The earth's gravity field is often represented as a spherical harmonic series, which is constructed using measured data. Much of the data used for this purpose can be obtained from satellite altimetry; the high-frequency part (being very localized) usually requires surface surveys. Computation of gravity at altitude, such as in a trajectory model, can be accomplished in several ways. In guidance, and other simple trajectory models, the simple inverse square equation is often used. This can be extended to include the oblateness effects. Higher fidelity models, such as those needed for D5 fire control, make use of the higher degree and order terms in the spherical harmonic series and the measured data.

Gravity at altitude is computed from a Stoke's integral formulation using a process known as ***upward continuation***. In theory, this process requires the integration of gravity effects over the entire earth to compute gravity at a single point in space. As a practical matter, the value of gravity at a point in space is more dependent on surface gravity at points in close proximity to a position directly under it. Dahlgren developed a kernel for the Stoke's integral that uses only the required points and a unique circular template of gravity data for use in the integration. The template, which is constructed in fire control, is centered at the point on the earth under the point in space and combines gravity data from different fidelity databases in an optimum fashion. This unique, and now widely recognized, result was a key determinant in achieving the required D5 accuracy.

Mark 98 Mod 1 Fire Control

The Mark 98 Mod 1 FCS was developed for TRIDENT II. A number of changes were made to ensure that readiness time and firing rate were maintained; the architecture is illustrated in Figure 4. Some significant changes include increasing fire control memory, replacing the disk packs with a new fixed mass memory device, adding high-density magnetic tapes to transport programs and data to the submarine, and adding digital interfaces (with microprocessors) between fire control and guidance (GISS) and fire control and navigation (NISS). Dahlgren participated in identifying system requirements and in developing the architecture.

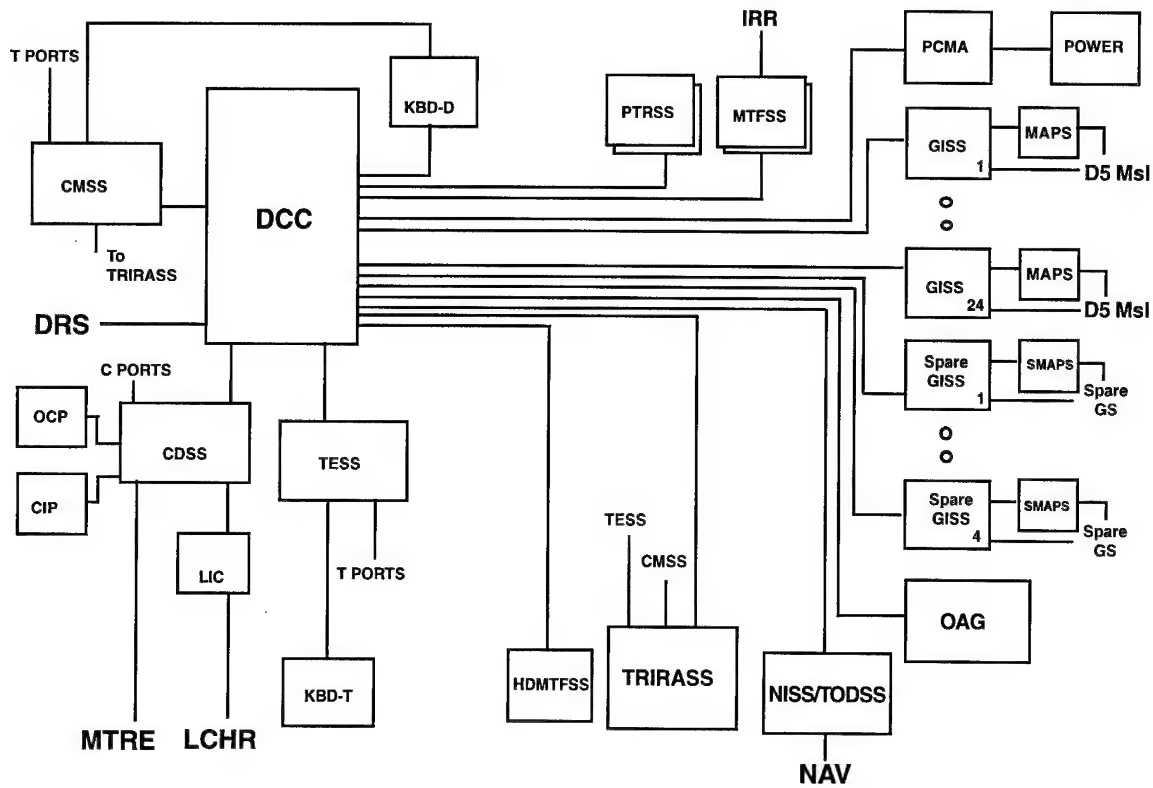


Figure 4. *Mark 98 Mod 1 architecture.*

The Future

The SLBM program has evolved in response to changes in the threat and in national policy. In 1959, the need was for a survivable system that supported the national policy of mutually assured destruction by providing a guaranteed retaliatory capability. Precise accuracy was not needed, nor was it achievable with the technology of the day. Initial upgrades to POLARIS were motivated by the need to increase range in order to provide greater submarine operating area and, hence, survivability. Later on, changes were made to accept new reentry bodies (and more than one per missile). Coincident with this was a need to improve and maintain accuracy at ever longer ranges.

The SLBM system has also changed in response to other needs. The U.S. nuclear policy, for example, has changed—from massive retaliation in the 1950s to flexible response in the 1970s and beyond—and the later SLBM system capabilities reflect this. Fire control changes reflect the increased need for targeting flexibility as well as the changes to the missile and guidance systems highlighted previously. Arms control agreements (from SALT to START II) limit the number and types of strategic systems available for nuclear deterrence. The reductions are inexorably moving the SLBM to a dominant position in the triad of nuclear deterrent forces. With this dominance comes the need for increased capabilities—such as D5's hard-target kill capability—and the need for flexible and responsive targeting.

What will motivate the development of future SLBM systems? Some would argue, based on the current world situation, that the need for nuclear deterrence is diminishing and that no system beyond D5 is needed. The recent Nuclear Posture Review established a continuing need for a D5 force, albeit with fewer Fleet Ballistic Missile Submarines (SSBNs), with the resultant need to "backfit" four of the original TRIDENT submarines so that they can operate with D5 missiles. Thus, increasing the operating life of the current system becomes critical. Studies (such as the "Future Deterrence Study" sponsored by the Strategy and Policy Division of the Office of the Chief of Naval Operations)[9] have considered the implications of the changing world situation and a range of threats

possible in the future. Nearly all of them suggest a move away from the bipolar world that has characterized the Cold-War era to a world with a number of smaller nuclear-capable countries. They also tend to suggest that providing strategic deterrence in the future will require more than nuclear weapons. Conventionally armed submarine launched ballistic missiles (CSLBM) have been proposed as a means to meet these needs.

It could be argued that, in this environment, fire control upgrades are more important than ever. Inherent in increased targeting flexibility, for example, is a need for improved fire control capabilities. Implicit in life extension are increased operating life and supportability for shipboard systems. Any new warhead, and especially a conventional one, will require changes to fire control. NSWCDD is actively supporting SSP by providing solutions to these problems and by preparing special revisions of software to support flight tests of the new capabilities. Targeting upgrades are being addressed by the SLBM Retargeting System (SRS). This has resulted in both shore-based processing improvements at NSWCDD and changes in shipboard fire control.

Some shipboard systems pose long-term supportability concerns. The changes that will be made to the Mark 98 Mod 1 FCS architecture by 1996 (Figure 5) address these concerns. A comparison with Figure 4 highlights the major changes: replacement of the mass storage system and connectivity with the SSBN's integrated radio room (IRR) using magnetic tapes. The D5 Mass Memory Subsystem (MMSS) is the result of a joint effort between NSWCDD and Lockheed Martin Defense Systems (LMDS). It is an example of using industry standards and "off-the-shelf" components (the disk drive and optical drive units and the processor in the mass memory controller) in the SLBM system. Dahlgren established the subsystem requirements, participated in the evaluation of the hardware design (including processor selection), recommended the commercial computer language to be used, and developed an operating system kernel to interact with the TDCC as part of a distributed real-time system.

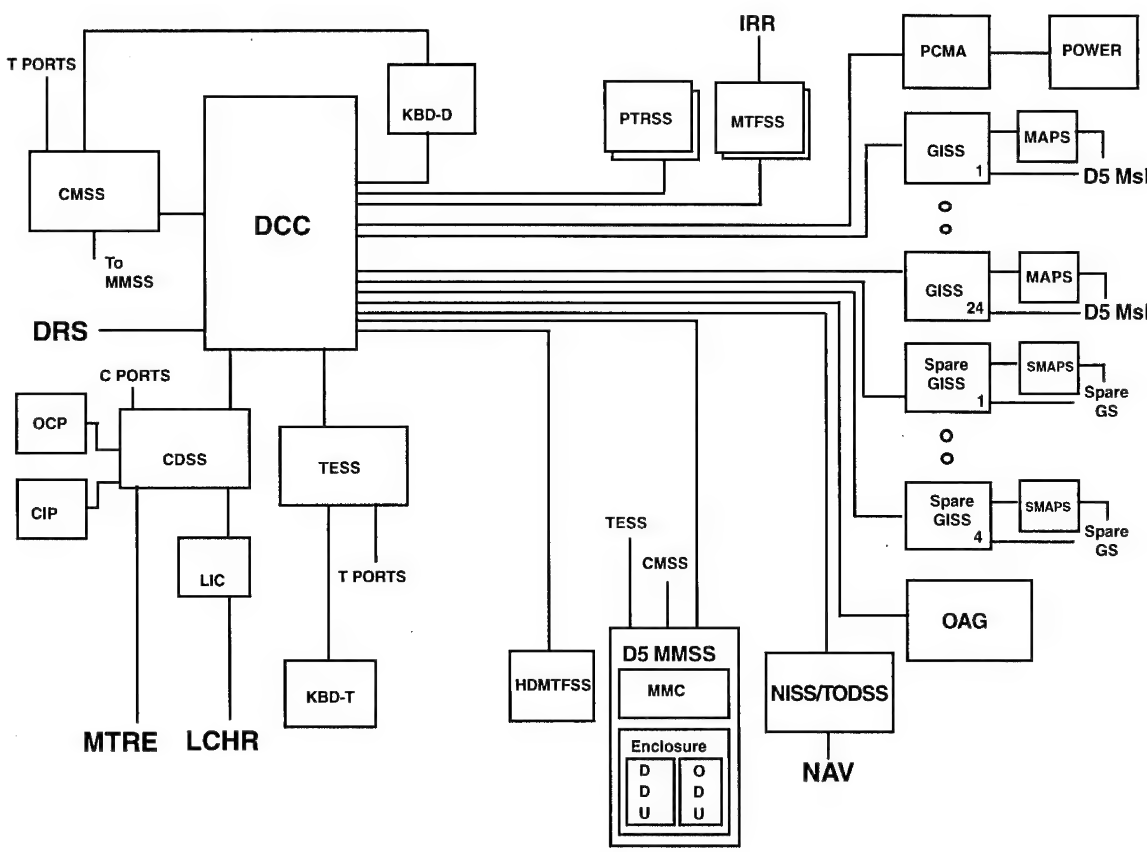


Figure 5. *Mark 98 Mod 1 architecture (1996).*

Longer term issues include upgrading the current FCS and addressing near- and far-term supportability concerns. The need for additional capabilities to address future requirements (targeting, for example) also drives redesign to meet future fire control needs and architectures. An example of a future architecture is given in Figure 6. This shows a fundamental change in architecture from the traditional computer-centered FCS to one that is fully distributed and which links all of the shipboard systems that make up the strategic weapons system. This new architecture reflects the changing nature of the SLBM fire control mission. Less obvious from this high-level view is that industry standard, off-the-shelf components (hardware, language, and operating system) will be used throughout the SLBM system. This architecture is flexible enough to support new requirements and reduce supportability costs, and is capable of being easily upgraded.

For nearly 40 years, NSWCDD has used its knowledge and experience in mathematics and computing to anticipate SLBM weapon system needs and to propose innovative fire control and targeting solutions. This effort is continuing and will help to ensure that the SLBM system can meet the changing requirements of its deterrent mission.

References

1. Brown, David R. Jr., *The Navy's High Road to Geoballistics, Part I*, Naval Surface Warfare Center, Dahlgren Division, Dec 1986.
2. McCollum, Kenneth G., ed., *Dahlgren*, Naval Surface Warfare Center, Dahlgren Division, Jun 1977.
3. Battin, Richard H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA, New York, NY, 1987.
4. Sapolsky, Harvey M., *The Polaris System Development: Bureaucratic and Programmatic Success in Government*, Harvard University Press, Cambridge, MA, 1972.
5. Fuhrman, R. A., "The Fleet Ballistic Missile System: Polaris to Trident," *Journal of Spacecraft and Rockets*, Vol. 15, No. 5, 1978.
6. Edwards, Donald T. and Daniels, Doreen H., *The SLBM Program: Brief History of NSWC's Involvement in FBM/SLBM Fire Control Systems, 1956-1984*, Naval Surface Warfare Center, Dahlgren Division, 1984.
7. Spinetti, Graham, *From Polaris to Trident: The Development of U.S. Fleet Ballistic Missile Technology*, Cambridge University Press, Cambridge, U.K., 1994.
8. Hughey, Raymond H., Jr. and Hill, Robert W., *Strategic and Space Systems*, Naval Surface Warfare Center, Dahlgren Division, 1985.
9. *Future Deterrence Study: Deterring the Use of Weapons of Mass Destruction*, Office of the Deputy Chief of Naval Operations for Plans, Policy, and Operations, Feb 1993.

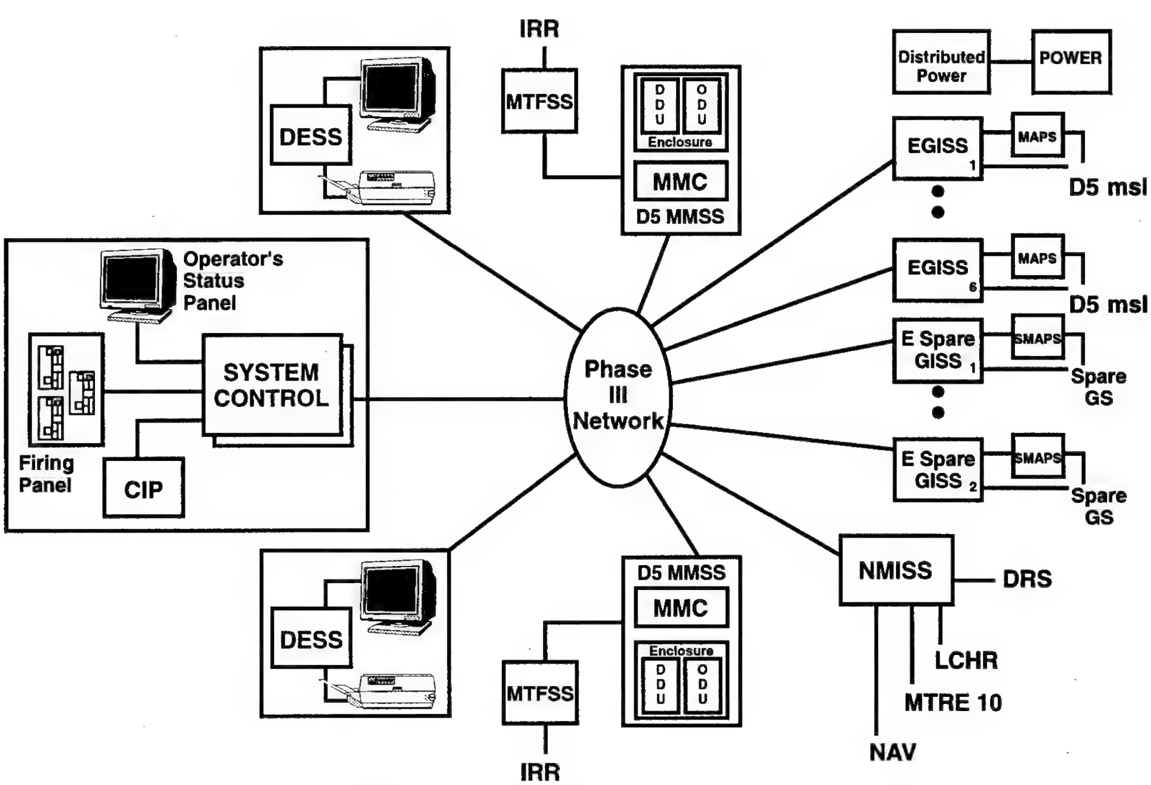
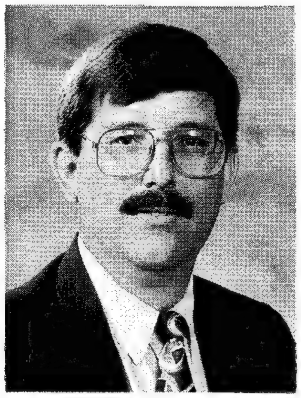


Figure 6. *Future fire control architecture.*

The Author



ROBERT V. GATES has a B.S. in physics from the Virginia Military Institute, an M. ENG. in engineering science from the Pennsylvania State University, an M.A. in political science from the Virginia Polytechnic Institute and State University, and has graduated with distinction from the Naval War College. He is a Principal Physicist in the Strategic and Space Systems Department of NSWCDD and is currently engaged in developing concepts for future Navy strategic systems, performing targeting analysis and support for the CNO staff, and providing leadership of the Strategic and Strike Warfare Focus Team at NSWCDD. Employed at NSWCDD since 1970, he has provided technical support in the areas of stellar inertial guidance, accuracy modeling and flight test support for the development of fire control and targeting software for TRIDENT I, TRIDENT II, and the United Kingdom POLARIS SLBM programs. He has also held a variety of management positions within the SLBM Research and Analysis Division at NSWCDD. He is a member of Pi Alpha Alpha, the national Public Administration Honor Society. Among his awards is the Navy Meritorious Civilian Service Award.

The Evolution of Strike Weapon Control

Alan Thomas, Andrew Horne, Carol A. Sheehan, and Warrington A. Tripp

The TOMAHAWK Weapon Control System (TWCS) provides information management, engagement planning, and launch control for TOMAHAWK cruise missiles on Navy surface combatants (see Figure 1). The computing architecture for this system has continually evolved to support new capabilities but has reached its performance and growth limits. To overcome these limitations and to respond to new requirements for strike weapon control and coordination, the Navy is developing the Advanced TWCS (ATWCS). The result is a flexible computing architecture with growth potential beyond TOMAHAWK to other strike weapons and new roles. This article traces the evolution of the TWCS, describes the new ATWCS architecture, and discusses challenges for the future.

Introduction

Weapon control is one of three elements that constitute the TOMAHAWK Weapon System, as illustrated in Figure 2. The most visible element, the TOMAHAWK cruise missile family, provides an attack capability for fixed land targets and ships. Currently, there are four types of TOMAHAWK missiles: the Conventional TOMAHAWK Land-Attack Missile (TLAM) with a unitary warhead (TLAM-C), the submunition dispensing version of the TLAM (TLAM-D), a nuclear land-attack version (TLAM-N), and the TOMAHAWK Antiship Missile (TASM). The nuclear and antiship missile variants are being phased out of the surface fleet.

A second element of the weapon system, TOMAHAWK mission planning, uses imagery and geographical information from national and tactical sources to plan and distribute TLAM missions, which comprised targets, overland routes, and employment information. Navy operators can plan missions at centralized shore sites, on aircraft carriers and selected command ships, and in relocatable vans. The mission-planning systems plan missions from a specified point to a target. Mission data can then be distributed to TOMAHAWK launch platforms (ships and submarines) on electronic media or via satellite communications links. Weapon control systems on the launch platforms can now plan the over-the-water missile route from their position to the first preplanned point.

Weapon control is the third component of the weapon system. The TWCS currently performs the TOMAHAWK weapon control function on surface ships, whereas the Combat Control System does this on submarines. The TWCS provides information management, engagement planning, and launch control via two types of launchers on four classes of ships. Virginia-class nuclear guided missile cruisers (CGN 38 class) and some Spruance-class destroyers (DD 963 class) use the AN/SWG-2 TWCS to launch TOMAHAWK missiles via two Armored Box Launchers (ABLs). The AN/SWG-3 TWCS found on AEGIS cruisers (CG 52 and above), most Spruance-class ships, and AEGIS guided missile destroyers (DDG 51 class) controls and launches missiles via the Mk 41 Vertical Launching System (VLS). Unlike the ABL, the VLS supports missiles other than TOMAHAWK.

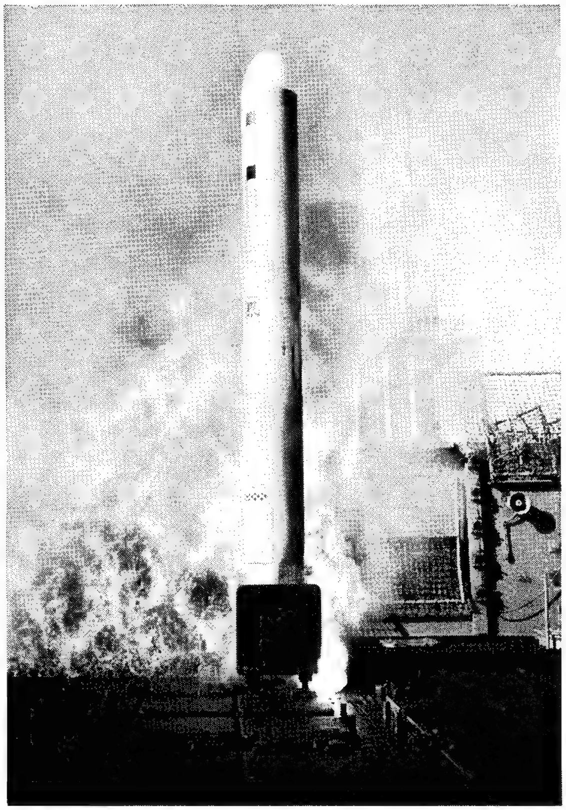


Figure 1. *TOMAHAWK firing.*

The TWCS provides digital interfaces to shipboard inertial navigation systems (INSs), launchers, combat control systems and offship communication systems. It is functionally partitioned into two subsystems as shown in Figure 3. The Track Control Group (TCG) receives and processes surface track data and electronic TLAM mission data updates, and performs engagement planning for the TLAM and the TASM. Engagement planning is the process of creating and evaluating the missile route from launch to the target for the antiship mission, or to the start of the preplanned mission for TLAM. The Launch Control Group (LCG) monitors missile and launcher status, conducts missile inventory operations, controls the launch sequence, and provides critical interrupt functions via the VLS. The TOMAHAWK launch sequence requires:

- Missile selection
- Flight software and mission data loading
- Alignment of the missile guidance system
- Booster arming
- Fire orders

The computing architecture of the TWCS has continually evolved over the years to meet new requirements. This has culminated in the development of the ATWCS, which overcomes TWCS limitations and provides a modern, flexible architecture with growth potential. A vision of continuing evolution for the ATWCS provides new challenges for the future. This article will trace the evolution of the weapon control system from its origins and then briefly describe future challenges.

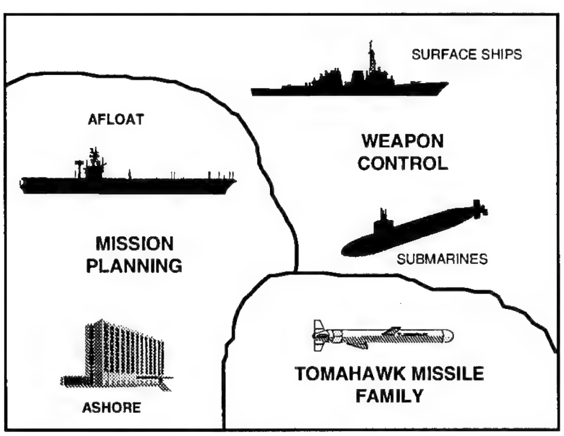


Figure 2. *TOMAHAWK weapon system.*

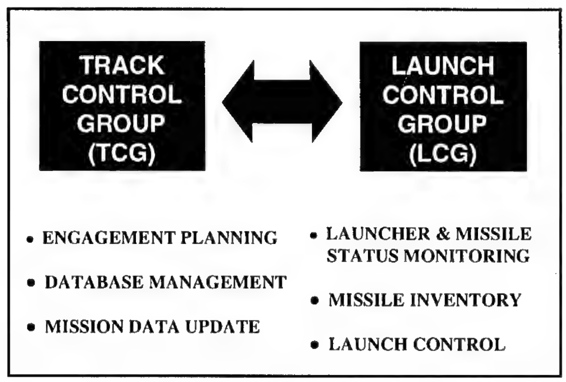


Figure 3. *Subsystems of the TWCS.*

Establishment of the TWCS

The TWCS evolved from a late 1970s' concept of a common weapon control system for the Navy surface launched cruise missile, the Air Force ground launched cruise missile, and the HARPOON antiship cruise missile. The Joint Cruise Missile Project Office, established in 1977 under the Naval Material Command, investigated hardware and software options for the weapon control system beyond the Navy computers available at the time. Rolm minicomputers were chosen for the system, primarily because they had

a flexible input/output structure (allowing specialized cards) and were not limited to 16 channels. In addition, the Rolm's memory could be expanded by adding an expansion chassis. A prototype system developed for the over-the-horizon targeting program *Outlaw Shark* used the Rolm computers and formed the basis for the TCG subsystem. This allowed reuse of much of the software. The LCG software was developed from scratch for these same Rolm computers.

The 16-bit Rolm computers were militarized commercial computers. These Complex Instruction Set Computers generally used over 200 basic instructions, provided 512 KB of memory, and executed 0.5 to 1 million instructions per second (MIPS), depending on the specific model. This is roughly equivalent to the processing power of a personal computer using an Intel 80286 processor. Ruggedized data storage was provided in the Random Access Storage System, adapted from a commercial product used in banking applications. It held two 65-MB removable magnetic disks called Data Transport Devices. These disk sizes are severely limited by today's standards. For example, 200 to 300 MB removable hard disks are now standard with laptop computers that incorporate fairly old technology.

There was an early emphasis on the Human-Computer Interface (HCI) for cruise missile weapon control. This led to selecting the Operator Interactive Display Terminal as the interface with operators. In addition to a 17-in. green monochrome cathode-ray tube , the terminal included fixed function keys, variable function keys on a plasma panel, a joystick, a keyboard, and a variety of audio alarms. It also provided some graphics capabilities for map and missile fly-out displays.

Developers used a functional design methodology for the TWCS software, which was constrained by the hardware, software tools, and standards of its day. FORTRAN 66 and Rolm Assembly Language were the primary programming languages used. Assembly Language provided fast execution where needed. The Rolm operating system was modified to support real-time operation.

The driving "real-time" requirements for weapon control were predictable timing (via software task prioritization) and the accurate time-delay estimations required for alignment of the TLAM guidance unit. The launch sequence requirements (e.g., tight timing on fire commands) and predictive algorithms for TASM engagement planning also stressed the computer performance. Because of these requirements and the state of technology at the time, some algorithms had to be embedded in interrupt routines to meet timing constraints.

For both technical and programmatic reasons, the surface- and ground-launched TOMAHAWK and the HARPOON programs diverged prior to deployment of a common weapon control system. Even so, some software and hardware remained in common after that point. The surface ship weapon control system was designated as the AN/SWG-2 TWCS and successfully reached initial operational capability in mid-1984.

The initially deployed TWCS, later designated Block 0, supported the ABL. The TCG used a Rolm 1666D computer, a Rolm 1602B computer, a data storage unit, and two operator terminals. The 1666D served as the main computer, while the 1602B computer served as a preprocessor for near-real-time track data. The LCG hardware consisted of a 1602B computer dedicated to each pair of launchers, a data storage unit, a 1666D computer as the main computer, and two terminals. Work on a version of the TWCS to support the vertical launch of TOMAHAWK missiles also began in the early 1980s, with USS *Norton Sound* serving as the test platform.

The Evolving TWCS

The TWCS was upgraded a number of times over nearly a decade to add new capabilities and to enhance system performance. The major upgrades to TWCS were designated as the Block I, Block II, and Block III upgrades. Each upgrade replaced the previous TWCS version in the fleet.

Block I

The Initial Operational Capability of the TLAM-D missile and TOMAHAWK on VLS-capable ships was realized from 1986 to 1988 with the Block I TWCS upgrade. The ABL configurations were also modified. Several changes brought increased software commonality among the systems that had been developed for the various launch platforms. The Block I

upgrade eliminated TCG configurations unique to battleships and VLS configurations, and it eliminated the unique LCG software for battleships (due to number of launchers). Addition of a new ship tracking algorithm, battle-group database management capabilities, test and training modes, new combat system interfaces, and other changes placed increased demands on the computing architecture.

On VLS-capable ships the Block I TCG consisted of three Rolm 1666B computers, two terminals, and a data storage unit. The TCG Primary Computer served as the main computer, while the TCG Secondary Computer (normally powered off) provided redundancy. The remaining Rolm 1666B computer was used as the track preprocessor. The ABL ships maintained the 1666D computers, while the TWCS software was upgraded on those platforms.

TCG interfaces to more powerful, external computers were necessary to support some requirements. An early example was a Hewlett Packard 9020 desktop computer that hosted algorithms to aid in engagement planning. This system was deployed as a stand-alone system at first, and was later interfaced to the TCG.

The Block I VLS LCG consisted of two Rolm 1666B computers, two operator terminals, and a data storage unit. As in the TCG, the LCG Primary Computer served as the main computer, while the Secondary Computer (normally on line) provided redundancy. These two computers were connected to two VLS computers and provided redundant data paths.

Block II

In 1989 the Block II upgrade introduced new capabilities and enhanced the existing functionality of the TWCS. Two significant new capabilities incorporated into the LCG were TLAM missile mission matching and nuclear flexible targeting. In the first, the LCG created a list of TLAMs in the ship's missile inventory sorted by how well the missile matched the requirements for a designated mission. Flexible targeting provided the launch platform with the capability to modify the terminal portion of an existing nuclear mission. Both of these compounded the LCG processing burden.

Interface management had placed an excessive burden on the TCG. A more powerful, adjunct computer was required to alleviate this problem and meet new requirements, such as the processing of electronic intelligence reports. Much of the communications processing was moved to an Asynchronous Communications Interface Card based on the Motorola 68000 microprocessor. The software for the card was written in the "C" programming language, which introduced a more modern language to the TWCS.

ABL ships received a hardware upgrade to maximize commonality across TWCS platforms and to support the new requirements. The older computers used in the ABL system were replaced by Rolm 1666B computers. Addition of a Secondary Computer for TCG added a level of redundancy. These upgrades nearly doubled the processing power of TWCS on ABL platforms.

The memory in the TCG Primary and Secondary Computers for both VLS and ABL ships was upgraded from a 512 KB two-way interleaved memory to a 1024 KB four-way interleaved memory. This upgrade enhanced performance by reducing memory access time and the number of disc accesses.

Block III

The TWCS was next upgraded to support time-on-target control of the Block III TLAM-C/D. This required calculation of the launch time necessary to achieve a desired time on target by modeling missile flight. The launch time had to be recalculated periodically to account for changes in ship position and heading. There were additional periodic calculations to determine the feasibility of achieving the desired launch position and desired time on target. A similar enhancement for TASM was also incorporated in the TWCS. It required periodic calculation of the launch time needed for achieving the desired time to activate the TASM seeker, as well as feasibility checks. These calculations increased the computational burden on the TCG.

An interface between the TCG Primary Computer and the TCG Secondary Computer was established to handle the increased processing requirements. The computations for the feasibility tests and for the TASM launch time calculation were performed in the TCG Secondary Computer. If the TCG Secondary Computer

failed, then all computations were performed in the TCG Primary Computer, resulting in degraded system performance.

LCG was enhanced to update computer programs stored in two TLAM subsystems: the new missile subsystem that processes Global Positioning System (GPS) updates and an upgraded Digital Scene Matching Area Correlator Unit. To support these and other enhancements to launch control, the memory in the LCG Primary and Secondary Computers was upgraded to a 1024 KB four-way interleaved memory. This upgrade reduced average memory-access time and reduced the number of disk accesses, thereby increasing system throughput. A representative Block III TWCS configuration is shown in Figure 4.

TWCS Reaching its Limits

In the Block II timeframe, it became clear that the TWCS' computing architecture, particularly the TCG subsystem, was reaching its limits. Future improvements were constrained by the aging technology and system architecture. The Block III upgrade and subsequent enhancements later reinforced this conclusion. In 1989, PMA 282—the TWCS program manager in the Naval Air Systems Command—initiated a study to provide system recommendations to support future improvements.[1] The final study recommended a complete upgrade to the TCG computing architecture. This included replacement of the storage devices; Rolm computers; and operator consoles with state-of-the-art workstations, embedded mass storage, network connectivity, and new software. This was later confirmed as the most effective choice by an independent Cost and Operational Effectiveness Analysis.[2]

In January 1991, millions of people worldwide saw TOMAHAWK put to the test in its first operational use during Operation Desert Storm. Many old employment paradigms, such as single-missile scenarios, were shattered. Fleet requirements for the TWCS were established to incorporate lessons learned, provide a more flexible and responsive weapons system, and accommodate new operational requirements.[3] These requirements formed the basis for the development of the ATWCS.

A new requirement for simultaneously preparing 32 missiles strained the processing power of the LCG hardware. Since plans were

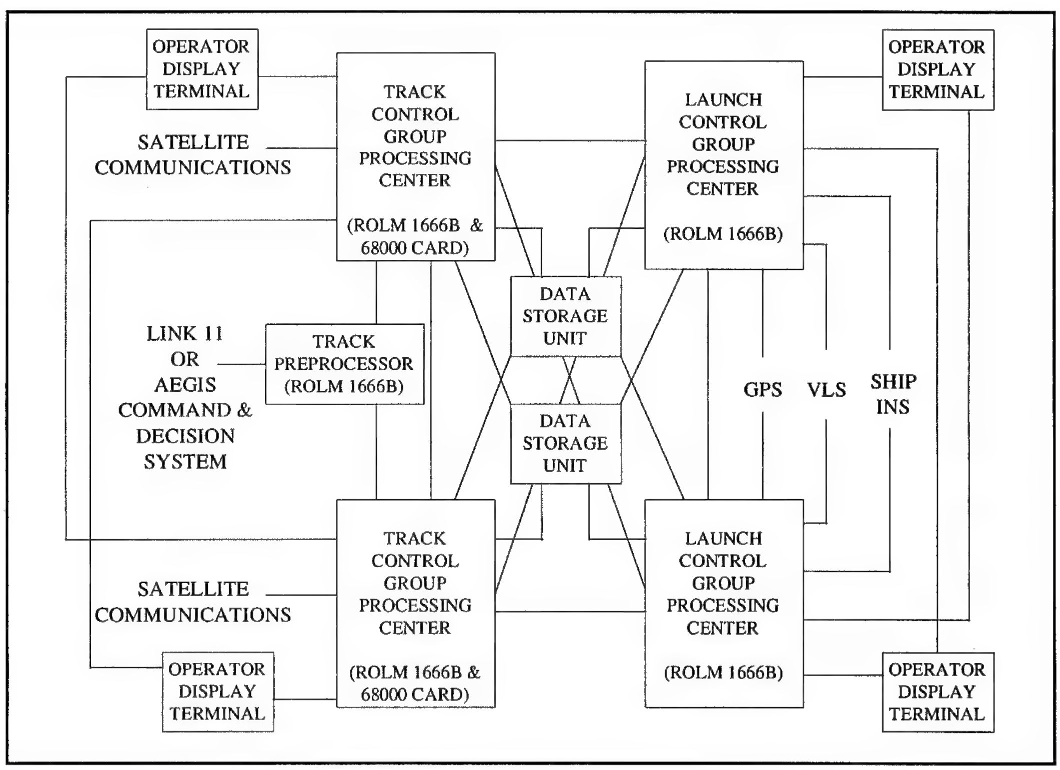


Figure 4. *Representative Block III TWCS configuration.*

preassembled during run time, storage for 32 plans was allocated to the system disk. With increased land-attack mission size and requirements for updating the missile flight software, the LCG processing capability was stretched to its limits in meeting the timing requirements. Additional capabilities could not be added without affecting the launch sequence time line. Plans for a Block IV TOMAHAWK missile brought additional attention to the aging architecture of the LCG. This resulted in a requirement to replace the LCG subsystem.

While ATWCS concepts were being formulated, plans were being made for a "Post Block III" upgrade to the TWCS. The resulting enhancements incorporated some lessons learned from Desert Storm, launch control support for ATWCS phased development, and AEGIS Combat System upgrades. Particularly significant were the new ability to prepare "backup" TLAMs for use when missiles failed and the ability to change over-the-water missile routes late in a launch sequence. This last upgrade to the TWCS gave the fleet new capabilities while the ATWCS was being designed and developed.

The Advanced TWCS (ATWCS)

The Program Management Proposal for ATWCS was approved by RADM G. Wagner, the Program Executive Officer for Cruise Missiles Project and Unmanned Aerial Vehicles Project (PEO(CU)) and by the Pentagon sponsor in February 1992, and then by the Assistant Secretary of the Navy for Research, Development, and Acquisition in July 1992. December 1992 brought approval to move to an engineering, manufacturing, and development phase. An Operational Requirements Document for ATWCS was signed in July 1994.

Significant upheaval in world politics helped mold the ATWCS concept of operations. The collapse of the old Soviet empire allowed the Department of Defense (DoD) to reduce its emphasis on nuclear weapons and the long-range, offensive antiship attack mission. The result was that ATWCS is not required to support either TLAM-N or TASM. The removal of the nuclear requirement was especially significant, because it allowed more flexibility in designing the ATWCS architecture. Both changes in emphasis had a significant effect on the operational concept.

The operational concept for ATWCS is intended to satisfy user needs documented in a fleet-generated Mission Needs Paper[3] and an Operational Requirements Document.[4] These needs include an improved operator interface to reduce operator workload, an improved ability to train operators and support strike coordination, and the capability to modify existing land-attack missions. The primary mission of ATWCS is to prepare and launch conventional TOMAHAWK missiles, which requires the engagement planning, track database management, mission update, and launch control functions. The secondary mission is to support strike coordination and battle group database management and to provide enhanced operator support via embedded training, on-line help and documentation, and maintenance support.

The ATWCS is being developed in two phases as shown in Figure 5. Phase I provides a replacement for the TCG subsystem. ATWCS Phase II will complete the new system architecture by replacing the LCG subsystem. First, the development approach for ATWCS will be discussed; then, the two phases of ATWCS will be described.

Development Approach

In 1991 the Naval Research Advisory Committee (NRAC) conducted a study on Open Systems Architecture for Navy command, control, communications, computers, and intelligence (C^4I) systems.[5] NRAC concluded that the Navy's adoption of an Open Systems Architecture philosophy would reduce development time and system cost while promoting the incorporation of new technology. The committee determined that these benefits also apply to combat systems. NRAC recommended the maximum use of industry standards, as well as commercial off-the-shelf (COTS) products, and emphasized a high degree of module/system reusability, maintainability, testability, understandability, portability, and affordability.

PMA-282 adopted this open-systems approach for ATWCS. The ATWCS architecture was designed using the layering concept derived from the International Standards Organization's Open System Interconnect

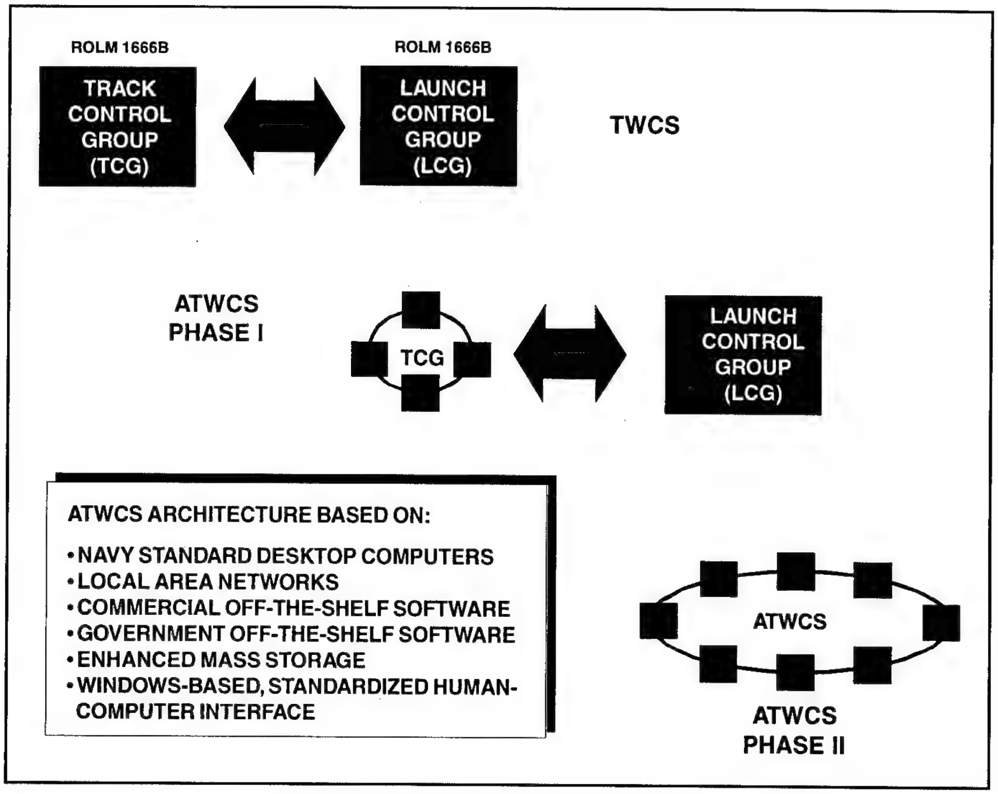


Figure 5. *ATWCS phased development.*

Model (an interconnection standard), which is depicted in Figure 6. This allows the layering of commercially available hardware and software, and the unique application software required to provide the functionality in the ATWCS. Five design agents are responsible for the design of the ATWCS software: Lockheed-Martin Marietta Austin Operations, McDonnell Douglas Aerospace; Naval Surface Warfare Center, Dahlgren Division (NSWCDD); Southeastern Computer Consultants Inc.; and Tiburon.

Whenever possible, ATWCS utilizes COTS hardware and software, including nondevelopmental hardware items. Workstations developed

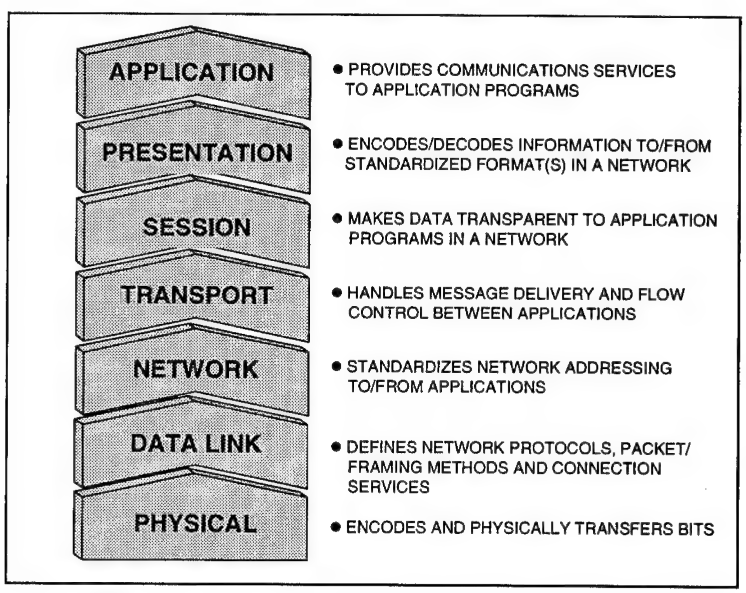


Figure 6. *Open-systems interconnect reference model.*

under the Navy Standard Tactical Computer program and other nondevelopmental items are the basis for the ATWCS architecture. The initial workstation is a Hewlett Packard Apollo Model 755, which is based on a Reduced Instruction Set Computer processor. This processor runs at 99 MHZ and is capable of executing 124 MIPS or 40 million floating point operations per second. Large random access memory (RAM) and disk capacities measured in gigabytes are provided with each workstation. ATWCS will adopt even more capable workstations as new commercial technology is incorporated into the Navy standard tactical computer program.

ATWCS development is also maximizing the use of industry standards, as well as COTS and government off-the-shelf (GOTS) software packages. COTS and GOTS will allow ATWCS to vastly improve the architecture and the HCI of the system. In addition, use of the Joint Maritime Command Information System (JMCIS) Common Operating Environment promotes commonality with command, control, communications, computers, and intelligence systems. The use of the Ada programming language has also helped to provide a robust and maintainable software architecture.

The ATWCS hardware architecture provides a distributed processing environment that will be exploited by a client-server software architecture. An "any workstation, any function" approach is being taken to ensure that the system can be easily reconfigured as necessary. This will improve overall system reliability by supporting gracefully degraded operation in the event of a hardware failure.

ATWCS software developers are incorporating solid software engineering practices in its design. The ATWCS application software is based on a layering concept, which provides a means of reducing software dependencies and provides a system that is more extensible and portable. Figure 7 depicts the three ATWCS software layers. The base layer contains the services that are provided by COTS products, such as operating systems and database management. The support layer provides ATWCS utility services that are used by the applications. The applications layer provides the high level functions that perform the primary ATWCS missions. Communication among the layers is facilitated by Application Programming Interfaces (APIs). APIs allow applications to use and provide services at a procedure call level, thus hiding the low-level

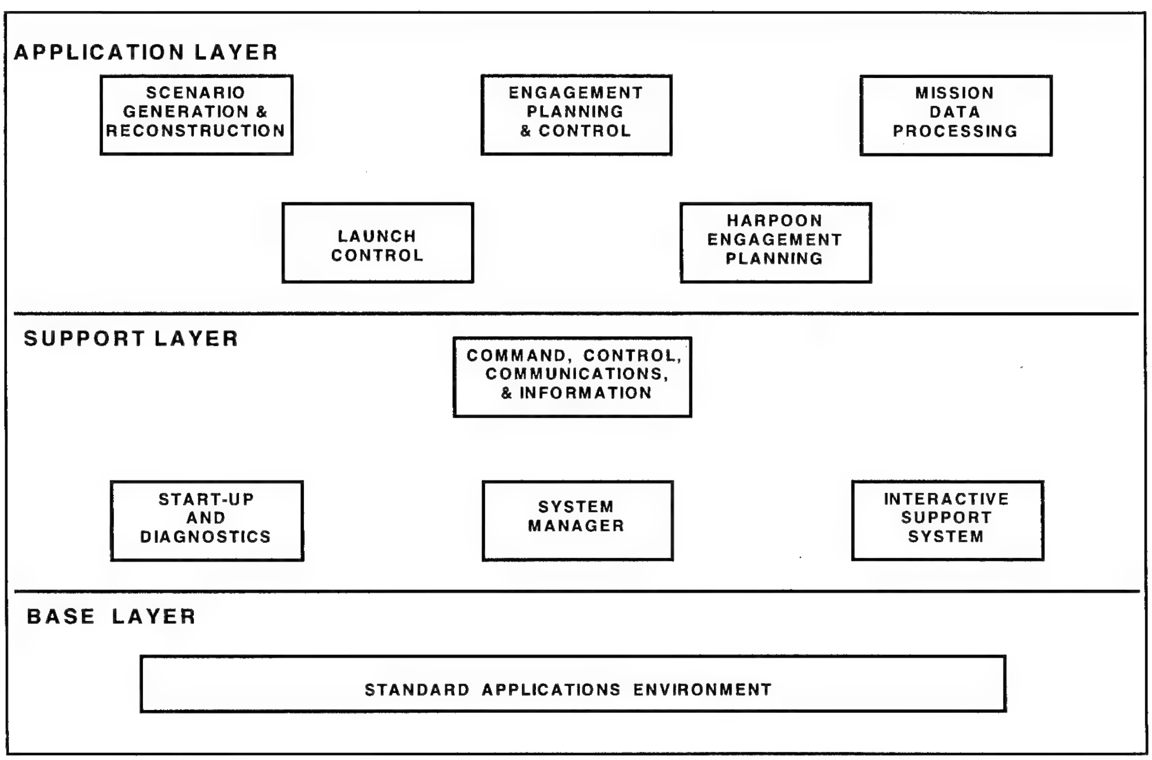


Figure 7. *ATWCS software layers.*

implementation. This helps mitigate the risk associated with the integration of large software systems.

The ATWCS software architecture allows ATWCS developers to take advantage of many modern software engineering methodologies and tools. Preliminary software size-scoping for ATWCS indicated that decomposition in the software design was required to manage its complexity. The software has been broken down into Computer Software Configuration Items (CSCIs). Many of the ATWCS design agents have incorporated the benefits of an object-oriented methodology in order to manage both the decomposition and development of their respective CSCI. Use of an object-oriented approach encourages good software engineering practices (information hiding, data abstraction, and encapsulation) and facilitates reusability, maintainability, portability, and extendibility. In this approach, objects or classes of objects are used as the basic building blocks of a system. Object attributes and the services that manipulate these attributes are both encapsulated within an object. This methodology also allows for a consistent representation across life-cycle phases: object-oriented analysis results flow into object-oriented design representations which are, in turn, implemented in objected-oriented programming constructs.

The development of ATWCS has given the software developers an opportunity to enhance their development environments with a commercial toolset that improves productivity and improves and automates software development. For example, ATWCS developers can take advantage of commercial user interface management systems and graphical user interface tools that can be used to produce software more efficiently than was possible when writing user interface code from scratch. Paradigm Plus, a computer-aided software engineering tool, is used in object-oriented analysis and design, as well as document production. The McCabe toolset is being used to aid in test development and complexity analysis. Ada software quality is analyzed using AdaMAT, another commercial tool. Use of these tools will allow developers to identify potential problems in the software architecture early in the system life cycle.

ATWCS Phase I

ATWCS Phase I, which will reach its initial operational capability in FY 96, replaces the TCG subsystem with a networked suite of two Tactical Advanced Computer (TAC)-3 workstations and four additional TAC-3 computers. In addition, the two LCG storage devices are replaced by a File Server Control Center, which includes a redundant TAC-3 computer. Industry standard Local Area Networks (LANs) are used as the primary means of communication between workstations. There are also point-to-point interfaces within the system for communication with older, external systems.

In addition to fully replacing TCG functionality, ATWCS Phase I provides strike coordination tools, TLAM mission display, and access to intelligence databases. Much of the new capability comes through the use of GOTS software. For example, a subset of the functions in the JMCIS is embedded into the ATWCS environment by use of the JMCIS Common Operating Environment. This software, the same core used in JMCIS, provides a track correlation and database management capability, and the basic HCI. Likewise, the Mission Data System software, used by TOMAHAWK mission planning systems to display TLAM mission data, is hosted on ATWCS and can be accessed via a window. This provides access to TLAM mission data and strike coordination tools.

The ATWCS Phase I architecture is also being adapted for SSN-688 class submarines, which carry TOMAHAWK missiles. ATWCS will provide engagement planning, and track database management, as well as provide mission data display, on-line help, and embedded training functionality for these submarines. The submarine's Combat Control System will continue to provide launch control functionality for TOMAHAWK missiles. This "Submarine ATWCS" upgrade will provide a higher degree of commonality between submarines and surface ships.

ATWCS Phase II

ATWCS Phase II completes the new architecture for the weapon control system and

supports future TOMAHAWK missile requirements. The Full Operational Capability for ATWCS Phase II is scheduled for FY 98. The system architecture is shown in Figure 8. In October 1992, NSWCDD conducted a prototyping effort to mitigate the technical risks of hosting the launch control functionality in the ATWCS hardware and software environment. Two significant issues addressed by this effort were multilevel security and TLAM guidance alignment requirements.

Physical separation is used in the ATWCS architecture to support multilevel security requirements. There are two primary LANs in the ATWCS architecture: the Weapon Control LAN and the Mission Data LAN. Two more LANs provide connectivity with JMCIS and with real-time computers, which will be discussed further on in this article. TAC computers on the Weapon Control and Mission Data LANs communicate via point-to-point interfaces.

TLAM guidance alignment requirements were also addressed during the Phase II prototyping efforts. It was determined that the TAC-3 processing was fast enough to calculate and send alignment data to the missile. However, the TAC-3 could not consistently time tag the data with the required 5.1-ms accuracy. This problem was attributed to the nondeterministic manner in which the UNIX operating system schedules processes.

ATWCS system engineers determined that a new architecture was needed to meet the requirements for time-tagging the alignment data. A single-board computer, the Hewlett Packard 742 Real-Time Computer, running the HP-RT real-time operating system and available on the standard Navy TAC contract, was chosen for the prototyping. This configuration allowed the developers to deterministically schedule software processes and, thus, be able to consistently and accurately time-tag the alignment data.

During alignment processing, the Real-Time Computer receives the ship's inertial navigation data, calculates and time tags the TLAM alignment data, and sends the results to the VLS for transmission to the missile. To provide redundancy there are two Real-Time Computers in the Phase II architecture. It was found that this real-time extension to the ATWCS architecture performed much better than required. The result of the prototyping effort was a robust, real-time processing capability with growth potential.

ATWCS Phase II will also provide an improved HCI and automated engagement

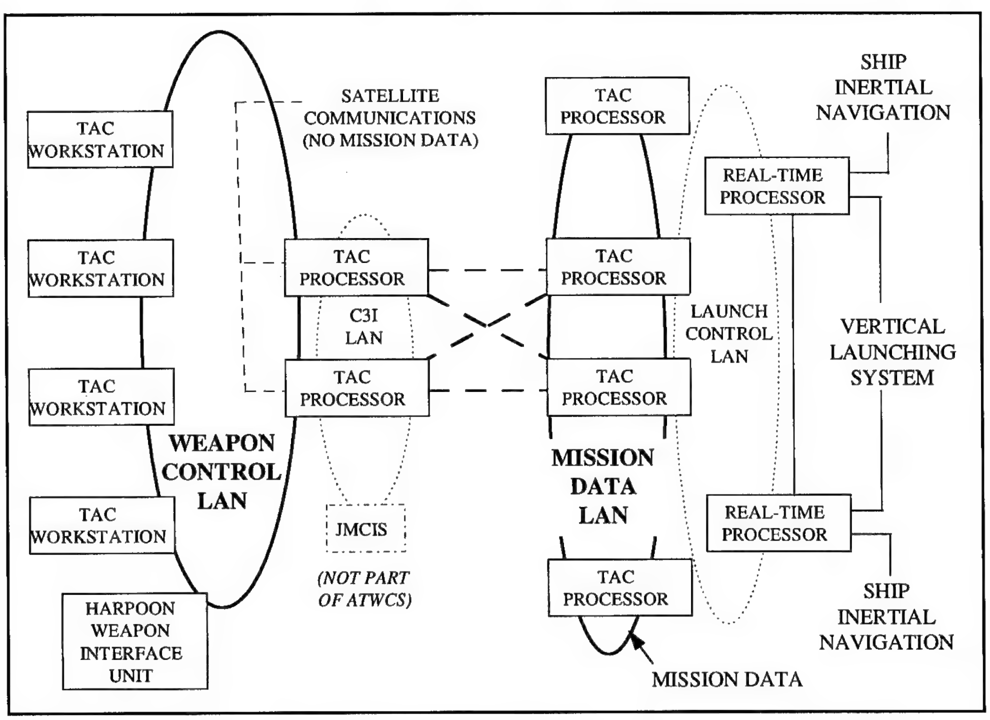


Figure 8. *ATWCS PHASE II hardware architecture.*

planning for HARPOON antiship missiles. This upgrade will also allow the replacement of outdated, specialized hardware and will standardize similar functions performed for both TOMAHAWK and HARPOON. From a systems perspective, it begins to extend the ATWCS architecture to strike weapons other than TOMAHAWK missiles.

The ATWCS hardware and software architecture will bring vastly increased processing power and enhanced memory, reduced cost, and a simplified, windows-based HCI. The new architecture also allows embedding Tactical Decision Aids and expanded training capabilities into the system. An architecture for real-time processing is being addressed in the Phase II development. The final result is a vastly improved system with an adaptable architecture, able to grow with new requirements.

Challenges Ahead

There are a number of challenges on the horizon for the ATWCS community. While the use of COTS products will provide significant benefits, it also presents near-term challenges. Configuration management will require a balance of controlling the introduction of COTS updates while replacing COTS no longer maintained by industry. This balance must be maintained while meeting ship schedules; the same is true of industry standards. To set targets for development, we must attempt to predict where the evolving standards are headed. In each case, the evaluation of both new products and new standards is crucial. The level of technical support provided, as well as the prospects for long-term support, licensing requirements, and the ability to influence future changes must be considered in the evaluation of new commercial products.

In the future there will be a continued Navy emphasis on striking fixed and relocatable land targets. The TOMAHAWK Baseline Improvement Program will result in the next-generation TOMAHAWK to enhance current Navy capabilities. This upgrade program will produce a Baseline IV TOMAHAWK Weapon System with improved accuracy, responsiveness, and target penetration using a common Block IV land and ship attack missile.

Automated mission planning for both the overwater and land missile routes will be provided on launch platforms. The shipboard system will monitor missile health and status in flight, support enroute changes in preloaded missions, and enable planning and in-flight updates for the antiship mission. Other capabilities, such as strike planning and coordination, must be upgraded to encompass the new warfare capabilities of Baseline IV. Lastly, a greatly reduced launch timeline will be provided by missile and ATWCS improvements. The result will be better Navy responsiveness to the joint air tasking cycle and the movement of some mission planning capabilities to surface combatants.

The next horizon towards which ATWCS should look is extending its architecture to support control of additional Navy power projection weapons. ATWCS Phase II provides a baseline in hardware, software, connectivity, and capability that enables this. The trend in projected Navy needs is towards a time-critical strike capability. This includes the growing Naval Surface Fire Support mission and response to mobile and emerging targets. Shipboard platforms must have near-real-time sensor connectivity, planning and targeting capabilities, and strike coordination and integration with new weapons. Changing the old paradigms first requires connectivity to near-real-time theater and tactical sensors, such as Unmanned Aerial Vehicles (UAVs), the Joint Surveillance and Target Attack Radar System (JSTARS) aircraft, and forces ashore. Integration of the Navy's prototype UAV Mission Planning and Control System into the ATWCS architecture is a first step toward the needed capabilities.

The ATWCS provides an effective architecture for integrating new power projection weapons into ships. Future surface strike weapons will come in several shapes and colors. They may include improved versions of the TOMAHAWK cruise missile, fast-response ballistic missiles, and naval guns firing extended range-guided

munitions. One proposed TOMAHAWK variant would deliver smart submunitions to GPS coordinates based on real-time JSTARS and UAV surveillance data. Real-time flight updates will enable the direct attack of or area denial to massed, mobile targets. This will require not only new connectivity, but also the retargeting of missiles in flight. Battle management—based on missile status, target movement, and battle damage indications—will present another challenge.

Fast flyout, ballistic missiles are also on the horizon to counter highly mobile and short-dwell land targets. The targeting algorithms and response times will be different than for cruise missiles, but similar targeting sensors and weapon control capabilities will be needed. The computing architecture that ATWCS employs will permit shipboard combat systems to accommodate such a missile and will enable better support of the Naval Surface Fire Support mission and quick response to time-critical targets during all phases of a conflict.

Next Generation Surface Combatants will introduce continuing challenges for ATWCS. Its role could be expanded even more, extending to targeting and mission planning support for guns and strike helicopters. Coordination of various power projection missions, such as Naval Surface Fire Support, will be critical. In addition, increased automation on a shipwide basis will be required to reduce manning requirements over today's ships. Multilevel security and systems integration will require particular attention. A shipwide computing architecture, common data structures, application program interfaces, and human computer interface will be needed. ATWCS has already made significant strides in all these areas.

It is clear that more shipboard functions will need to approach real time. Increased connectivity to joint assets and a Navy role in time-critical strike will drive more "jointness" into the system. Operationally, the Navy must address more rapid strike coordination and time-critical command and control. To adequately prosecute future targets sets, some level of control and coordination must be moved down to the surface combatants. Operational doctrine must evolve to take advantage of new technology—both commercial technology and the evolving joint technology base—to break the old paradigms.

Glossary

ABL: Armored Box Launcher
API: Application Program Interface
ATWCS: Advanced TOMAHAWK Weapon Control System
C^{4}I: Command, Control, Communications, Computers, and Intelligence
COTS: Commercial Off-The-Shelf
CSCI: Computer Software Configuration Item
DoD: Department of Defense
GPS: Global Positioning System
GOTS: Government Off-The-Shelf
HCI: Human-Computer Interface
INS: Inertial Navigation System
JMCIS: Joint Maritime Command Information System
JSTARS: Joint Surveillance and Target Attack Radar System
LAN: Local Area Network
LCG: Launch Control Group
MIPS: Millions of Instructions Per Second
NRAC: Naval Research Advisory Committee
PEO(CU): Program Executive Officer for Cruise Missiles Project and Unmanned Aerial Vehicles Project
RAM: Random Access Memory
TAC: Tactical Advanced Computer
TASM: TOMAHAWK Antiship Missile
TCG: Track Control Group
TLAM: TOMAHAWK Land-Attack Missile
TLAM-C: Conventional TOMAHAWK Land-Attack Missile
TLAM-D: Dispensing TOMAHAWK Land-Attack Missile (conventional)
TLAM-N: Nuclear TOMAHAWK Land-Attack Missile
TWCS: TOMAHAWK Weapon Control System
UAV: Unmanned Aerial Vehicle
VLS: Vertical Launching System

Acknowledgments

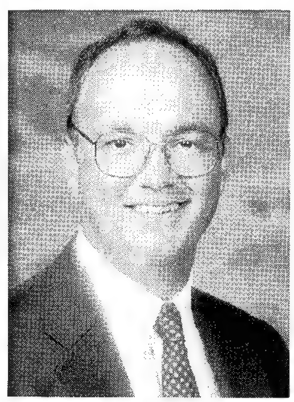
The authors wish to express their thanks to the Strike Systems Department personnel, too numerous to mention, who assisted invaluably in the writing of this paper. In particular, we wish to thank Mr. Wayne Harman, Leader for Strike Systems Concept Development and Advanced Technology, for his topic selection, initial ideas, and comments. Much information on the history of the TWCS was provided by Mr. Harman; Mr. John Straub, Chief TOMAHAWK Systems Engineer; Mr. Sonny Akin, former Deputy Program Manager for Mission Planning; Mr. Jim Blackwelder, Head of the Strike Weapon Systems Division; and Mr. David Daniel, acting Deputy PEO (CUO). We also wish to thank Mr. Richard Stutler, Head of the Strike Systems Planning Division, and Ms. Pat Yaroszewski, Head of the Weapons Software Engineering Branch, for their encouragement and review. Mr. Ron Pollard and Mr. Bernard Duren of the Combat Systems Department, and Mr. Kelly Blosser of the Warfare Analysis Department provided valuable insights regarding Next Generation Surface Combatants and the Naval Surface Fire Support mission.

References

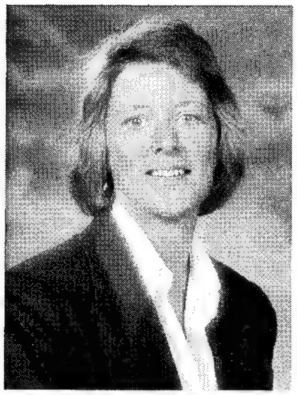
1. Johns Hopkins University/Applied Physics Laboratory, *Tomahawk Weapon System, Advanced Tomahawk Weapon Control System (ATWCS) Architecture Study*, FS-89-192, Dec 1989.
2. McPherson, Michael; Cardiel, Amanda; Rooker, Terry; Kennedy, Susan; and Waite, Steve, *ATWCS Cost and Operational Effectiveness Analysis (COEA)*, NSWCDD/TR-93/76, 17 Jun 1993.
3. *TOMAHAWK Weapon System (Advanced) Mission Needs Paper*, CINCLANTFLT Letter Ser N99/05287, 28 Jun 1991.
4. *Operational Requirements Document for Advanced TOMAHAWK Weapon Control System (ATWCS)*, 20 Apr 1994.
5. Office of the Assistant Secretary of the Navy (Research, Development and Acquisition), *1991 Summer Study on Open Systems for Command, Control, and Communications*, Naval Research Advisory Committee Report 91-1, Sep 1991.

The Authors

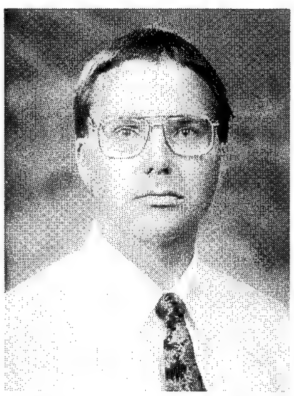
ALAN THOMAS graduated from the University of Virginia in 1985 with a B.A. degree in physics. He joined the TOMAHAWK System Test group that year, where he worked for five years. Mr. Thomas then led selected TWCS and ATWCS system requirements efforts for three years, particularly in the area of integration of TOMAHAWK into shipboard combat systems. Since 1993, he has been developing future concepts for strike systems. He has been the Strike Warfare Team Leader for the 21st Century Surface Combatant studies at NSWCDD, as well as the lead NSWCDD systems engineer for the "TOMAHAWK Stops the Attacking Regiments" (TSTAR) cruise missile concept. In addition, Mr. Thomas supports the Naval Surface Fire Support efforts at NSWCDD, with particular interest in near-real-time sensor integration, the shipboard integration of new power projection weapons, and the incorporation of new technology into shipboard systems.



ANDREW HORNE received a B.S. degree in computer science in 1982 from James Madison University in Harrisonburg, VA. Before joining the government in 1989, Mr. Horne worked for several years in industry supporting both the Trident missile and the TWCS. As the Group Leader of the ATWCS Launch Control Requirements and Verification Group, Mr. Horne took the lead in identifying, acquiring, and integrating cutting edge software development technologies and methodologies into the L Department development processes. Mr. Horne utilized his experience and knowledge in nuclear safety and tool development in providing critical support to the risk area identification and mitigation efforts of the ATWCS Phase II Software Development team. He recently assumed the position of Assistant Software Engineer for ATWCS, PMA-282T21.



CAROL A. SHEEHAN received a B.S. degree in computer science from Lock Haven University, Lock Haven, PA. Prior to joining NSWCDD in 1985, she worked at Brookhaven National Laboratory in Upton, NY. As a member of the Weapon Targeting Software Branch of the Strike Systems Department, she is the Independent Verification and Validation (IV&V) Group lead for the ATWCS software. Ms. Sheehan is an expert in the object-oriented analysis and design methodology, which was adopted for the ATWCS development effort, and was a lead in educating the ATWCS community in this methodology. Prior to assuming a lead IV&V role for ATWCS, Ms. Sheehan participated in the development of a prototype for replacing the TWCS Operator Interactive Display Terminal. Through this successful prototyping effort, the ATWCS program was conceived.



WARRINGTON A. TRIPP received a B.S. degree in applied mathematics in 1987 and an M.A. degree in mathematics in 1989, both from Western Michigan University. He received an M.S. degree in computer science from Virginia Polytechnic Institute and State University in December 1994. Mr. Tripp joined NSWCDD in 1989, where he has worked in software and system engineering for the TWCS and ATWCS. He has focused on algorithm development for TOMAHAWK engagement planning: ship movement tracking and prediction, TLAM time-on-target control, and TASM time-of-search calculations. Mr. Tripp leads the Mathematical Analysis Team supporting the IV&V tasking with the various ATWCS design agents. This team develops and/or validates algorithms in support of ATWCS requirements.

Mine Countermeasures Simulation

Elan Moritz

Mine Countermeasures (MCM), an element of Mine Warfare (MIW), is a critical component of the U.S. National Defense picture.[1] *MCM operations are mandatory since it is a simple and relatively inexpensive matter for opponents to place vast quantities of land and sea mines. MCM has historically been difficult and time-consuming to conduct. Representation of MCM and MIW activities in simulations can substantially accelerate and optimize material and tactical means development to address this problem. A systematic effort to simulate the breadth of MIW operations is described. Under the sponsorship of the Program Executive Officer, Mine Warfare, the significant progress and accomplishments with the Multiwarfare Analysis and Research System (MARS) are presented, including a discussion of the use of MARS(D)-Version 2.0 for the conduct of Simulation Exercise 95-1. Related efforts associated with MIW simulation, including an advanced simulation visualization program currently being coupled with MARS, are also described. Finally, challenges, issues, and vision for the use of modeling and simulation in an expeditionary warfare context are offered.*

Need for MCM Simulation

This paper discusses the evolving state of the art of MCM simulation. In particular, this paper focuses on advanced MCM constructive and virtual simulation technology developed at the Naval Surface Warfare Center, Dahlgren Division's (NSWCDD's) Coastal Systems Station (CSS).

MCM, an element of MIW, is a critical component of the U.S. national defense picture. MCM operations are mandatory since it is a simple and relatively inexpensive matter for opponents to place vast quantities of land and sea mines. The Department of the Navy is particularly concerned with mine threats in the littorals. As regards naval forces, mines located anywhere from deep water to the surf zones and beach regions pose formidable problems.

Operations to counter these mines are complex and time-consuming. The range of technologies involved span the phenomenology spectrum. These operations include covert and overt reconnaissance with active and passive sensors; command, control, communications, and information processing; manual mechanical electromagnetic and explosive neutralization; and highly sophisticated robotic systems and artificial intelligence manifested in unmanned and autonomous systems.

A key feature that must be continuously emphasized is the *immense complexity in MCM*. The complexity is generated by the diversity of mines and obstacles encountered as well as the enormous number of mines and obstacles one faces in real-life scenarios. Land mines and obstacles involved in a typical war number in the millions or tens of millions; the number of sea mines involved is in the tens to hundreds of thousands. To contrast the situation with other warfare areas, once placed, mines are typically immobile and silent; furthermore, their observable cross sections are low (due to their physical dimension) and many mines are placed in the most challenging physical environments to observe.

Effectors

By effectors, I mean systems used to effect a physical change; these include such systems as mechanical sweeps with explosive cutters, the mine neutralization system, and acoustic and magnetic sources that generate signals to sweep mines. Effectors stand in contrast to sensors, which are used for information gathering but do not attempt, of themselves, to cause a lasting physical change in their surroundings.

Adding to the complexity is the wide range and large number of platforms, sensor, and effector systems one must consider and utilize to successfully counter mines. These range from national reconnaissance assets to individual divers. The complexity is compounded by the different ranges, speeds, and time constants associated with the different systems; the generation of a common tactical picture; and the need for coherent asset management over periods ranging from days to weeks.

Specific needs associated with MCM (and MIW in general) fall into four user categories: war-fighter needs, senior policymaker needs, resource and acquisition manager needs, and developer (system designers and technologists) needs. The simulation-relevant technical needs can be grouped into war planning and plan evaluation, training, concept assessments, cost/effectiveness tradeoff analysis, and technical element design. Simulation offers the potential to overcome some of the complexities inherent in representing and mounting successful MCM operations, as well as the potential of rapidly obtaining critical insights for all decision stages associated with MCM acquisition and operations.

A Pedestrian Introduction to Mine Warfare

Mines have proven themselves effective weapons of choice in many armed conflicts. Mines are used to deny mobility and inflict severe damage to man and machine. MCM operations are extremely difficult in the best of circumstances; the inherent dependencies of sensors and effectors on environmental factors make MIW operations even more complex.

There are many types of mines. Fundamental to all mines are warheads (explosive charges) and actuation mechanisms. Warhead sizes range from ounces (antipersonnel (AP) mines), to pounds (antitank (AT) mines), through thousands of pounds (antiship (AS) mines, also termed seamines). Actuation mechanisms range from simple contact mechanisms to extremely sophisticated multisensor phenomenology with computerized signal processing targeting specific platforms via sophisticated signature matching.

The principal mine threat of concern to naval forces (shown in Figures 1 and 2) are those located in water-depth regions of deep water

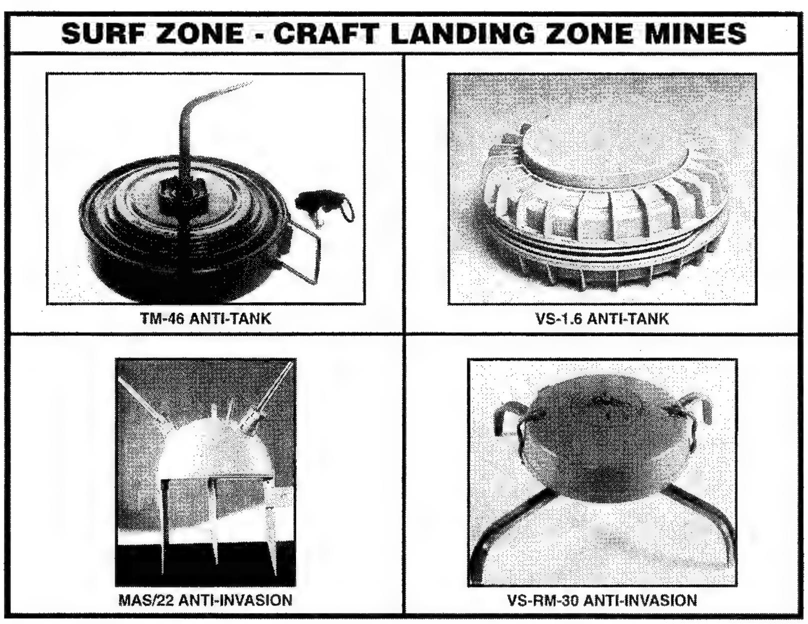


Figure 1. *Examples of surf-zone and beach/craft landing zone mines.*

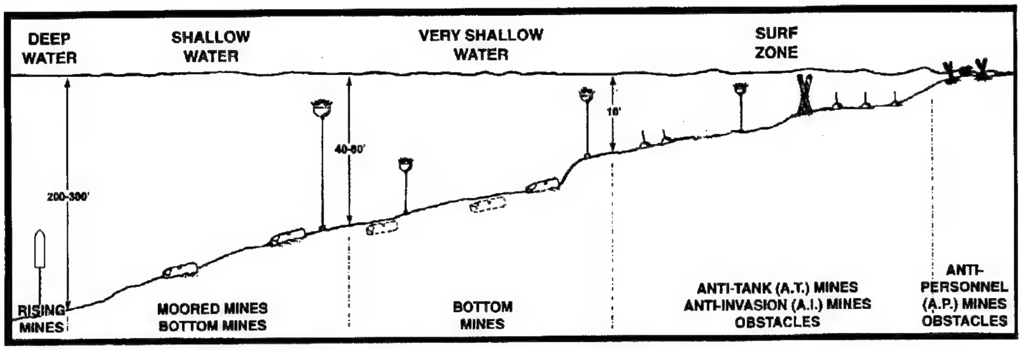


Figure 2. *Illustration of mine threat.*

(600 to 200 ft), shallow water (200 to 40 ft), very shallow water (40 to 10 ft), the surf zone (10 to 0 ft) and beach/craft landing zones (the respective acronyms are DW, SW, VSW, SZ, CLZ, and BZ). The warhead size and sophistication of the mines are correlated with water depths and the fact that deep-water mines target large platforms and possibly capital ships.

Deep-water mines cost upward of tens of thousands of dollars (although floating contact mines can be produced for a lower cost) and can seriously damage (or sink) billion-dollar capital ships. Surf and beach zone AT mines and antivehicle mines can cost in the tens of dollars. While antipersonnel mines can, and are likely to, be found (in quantities of millions and unit prices of dollars), the focus is on countering, and hence simulating, antitank/antivehicle mines and mines targeting larger platforms.

Beach/surf zone mines are typically placed on the sea bottom and may be environmentally or deliberately buried or partially buried. Larger mines can be placed on the sea-bottom; alternatively, mines can be moored with a short or long tether; some mines (typically simpler contact mines) may be cast off as floating mines. The advantage/disadvantage of these mines is that they are carried by currents and, hence, are hard to control.

In water depths greater than 10 ft, bottom mines are typically influence fuzed, while tethered mines may be either contact actuated or influence fuzed. Tethered mines are kept in place by tethers and anchors. In water 10-ft deep and shallower, bottom mines could be influence fuzed or contact based. Contact may be transduced through a variety of mechanisms such as long tilt rods, pressure plates, and actuation pins. To make them more sweep resistant, some mines are designed to sense and respond to multiple simultaneous influences (typically involving combinations of acoustic signatures, magnetic signatures, and pressure signatures.)

The generic approaches to countering mines include surveillance, reconnaissance, hunting, and identification (SRHI); avoidance; and clearance (which includes influence sweeping, mechanical sweeping, and explosive neutralization). A detailed hierarchical breakdown of MCM missions and tasks can be found in Reference 2 and is consistent with an overall Theater Mine Defense concept.[3]

Surveillance and reconnaissance may be conducted by both overhead assets and underwater assets. Overhead assets use sensors working in different parts of the electromagnetic spectrum, while underwater assets would use sensors that are mostly acoustic or seismic in nature (and possibly some magnetic sensors). Hunting and identification can utilize higher performance acoustic, magnetic, and electro-optic sensors. The platforms involved in SRHI run the gamut from fixed and deployed underwater sensor arrays; sensors mounted on manned, unmanned, and autonomous vehicles underwater as well as surface vehicles; manned and unmanned airbreathing airborne vehicles to satellites; and perhaps observation balloons and airships.

The kinds of information SRHI systems can collect include surveillance-based locations and status of mine production and depot facilities, movement of mines from these facilities to operational deployment units, tracks of deployers and actual mine (and obstacle) deployment events, reconnaissance of areas considered for naval operations, deliberate hunting for mines in very specific areas and, ultimately, identification of mines (perhaps to the level of make and model).

MCM operations, for the most part, are conducted by dedicated MCM forces that include MCM-1- and MHC-51-type surface MCM ships, MH53 helicopters, and specially trained Explosive Ordnance Disposal (EOD) and SEAL teams. The dedicated forces currently use a variety of systems (including surface ship or helicopter-towed sonars and mine neutralization systems (see Figure 3)). Key MCM function definitions and some functional allocations are described in Figures 4 and 5.

Degaussing and magnetic signature reductions are MCM functions intrinsically designed into most ships for self protection. Some platforms, such as Landing Craft Air Cushion (LCAC) (see Figure 6) and MCM-1, are designed for exceptionally low magnetic signatures.

Organizationally, dedicated MCM and mining operations come under the cognizance of Commander, Mine Warfare Command (CMWC). Operational units include MCM squadrons under the command of their respective group commanders. HM-14 and HM-15 are rapidly deployable Airborne MCM (AMCM) helicopter squadrons and are also part of the dedicated forces. HM-14 and HM-15 can be rapidly deployed to forward areas in time of critical needs. The AMCM platforms can use large-deck amphibious ships to stage operations.

An emerging element of MCM is the increased acceptance and demand for organic MCM (OMCM) capability. Inherent in OMCM is the precept that larger units such as Amphibious Ready Groups (ARGs), Carrier Battle Groups

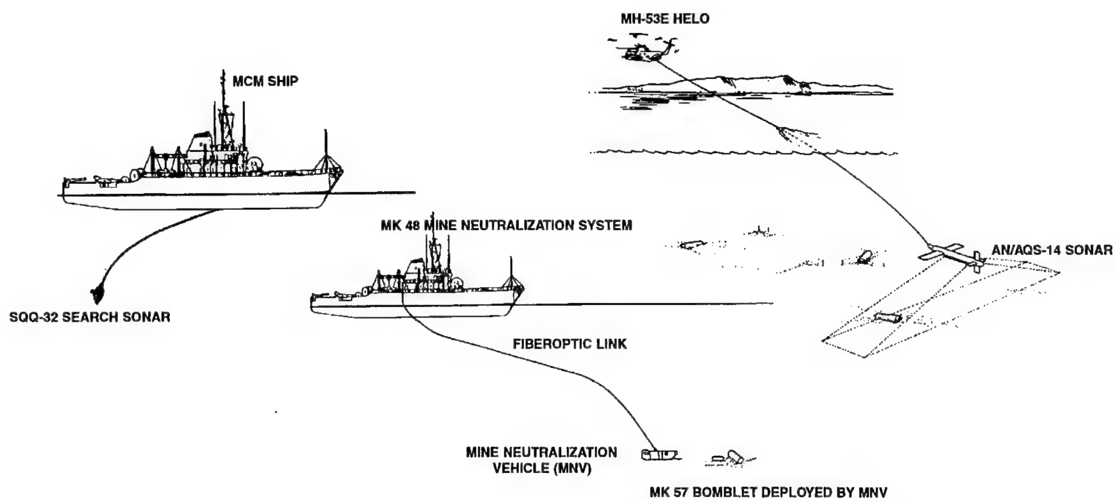


Figure 3. *Illustration of some MCM forces.*

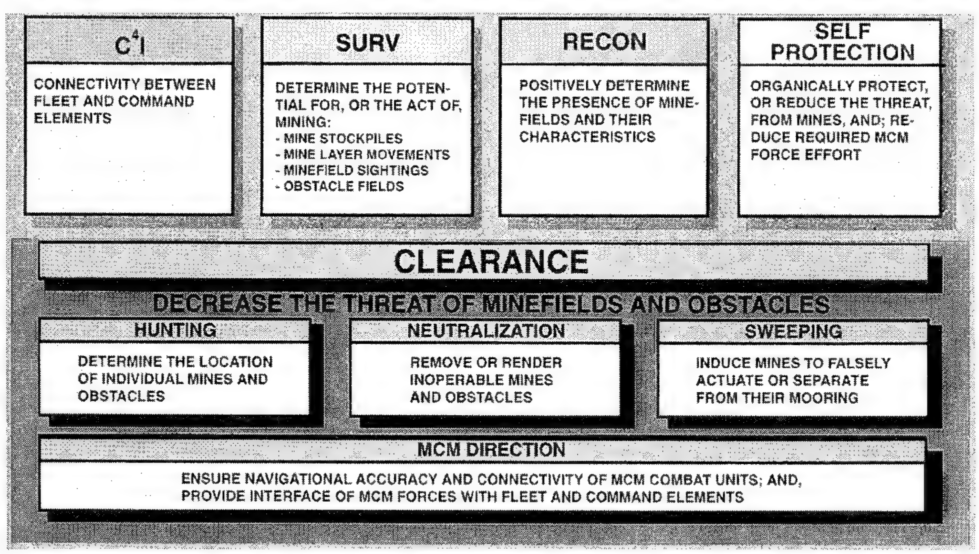


Figure 4. *Definition of MCM functions.*

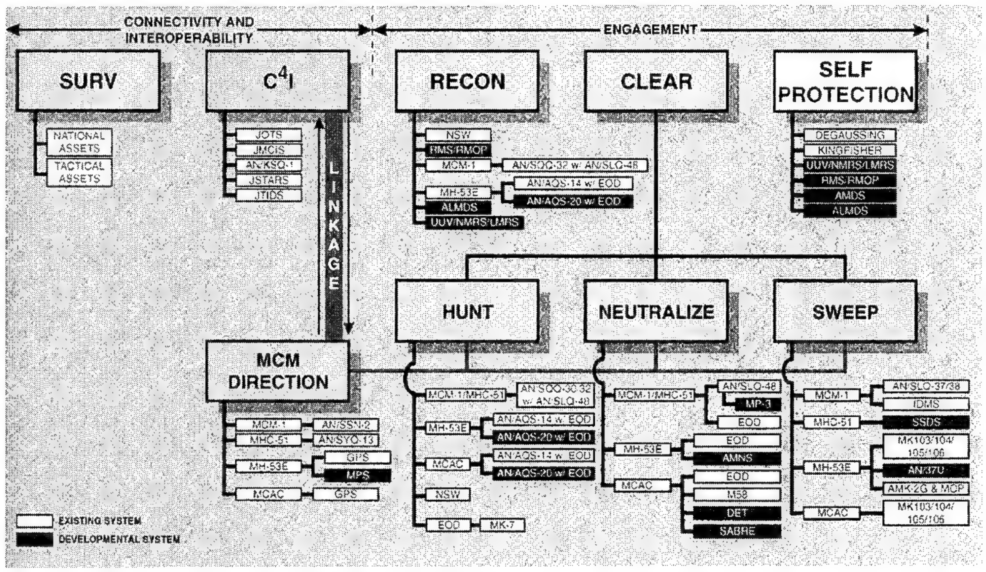


Figure 5. *MCM functional allocations.*

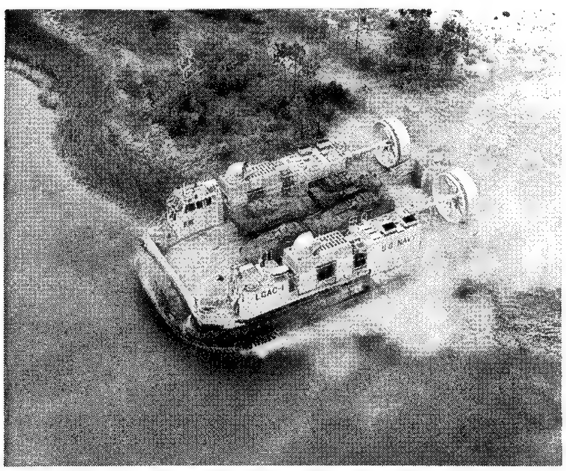


Figure 6. *Landing Craft Air Cushion (LCAC). LCAC can be used to transport heavy equipment such as tanks and light attack vehicles; a multipurpose version (MCAC) can carry MCM mine hunting and sweeping gear.*

(CVBGs), and Submarine Forces will carry MCM systems that at least allow these forces to avoid mined areas. Typically, such OMCM systems will be composed of sensor systems typified by the Magic Lantern electro-optic prototype, the remote mine hunter operational prototype (RMOP), and the near-term mine reconnaissance systems (NMRS) prototype (see Figure 7).

General Modeling and Simulation (M&S) Considerations

Modeling and simulation have been traditional tools in many technical disciplines for many years. Recently within the Department of Defense (DoD), special attention has been accorded to "Modeling and Simulation" (M&S) as a method for accelerating acquisition programs and training. This attention culminated in DoD Directive Number 5000.59 (which was issued January 4, 1994 and signed by then Undersecretary of Defense William Perry) calling for investments that:

> *. . . shall promote the enhancements of DoD M&S technologies in support of operational needs and the acquisition process; develop common tools, methodologies, and databases; and establish standards and protocols promoting the internetting, data exchange, open-system architecture, and software reusability of M&S applications.*

The directive calls for M&S applications:

> *to support the major DoD decision-making organizations and processes such as the Defense Planning and Resources Board; the Defense Acquisition Board; the Joint Requirements Oversight Council; and the DoD Planning, Programming, and Budgeting system.*

As a long-term objective, the directive also calls for interoperability, where models or simulations work with other models and simulations in order to enable them to operate effectively together.

The Department of the Navy recognized the benefits of M&S in a memorandum issued by

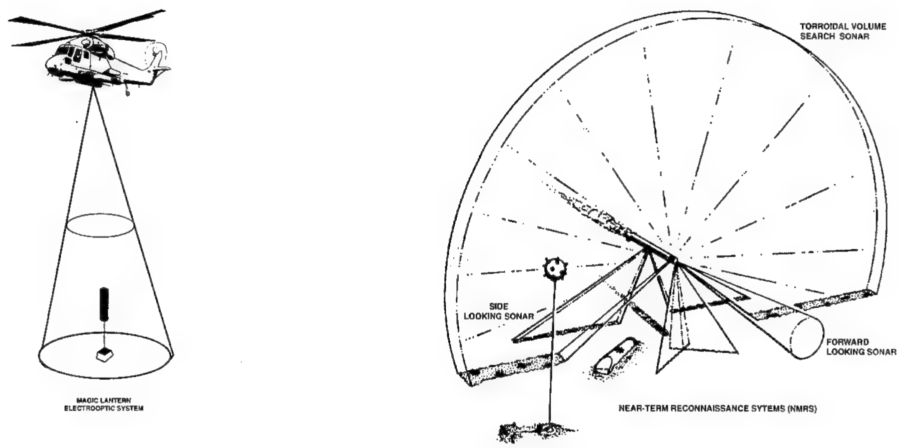


Figure 7. *Organic MCM (OMCM) sensor systems.*

the Assistant Secretary of the Navy.[4] This memorandum calls for expanded use of models and simulations to support all phases of milestone decisions of the acquisition cycle.

Since models and simulations have been used by many individuals in many contexts ever since the advent of the scientific method (and even earlier), it is best to use a consistent terminology. To this end, DoD 5000.59 offers key M&S definitions (which are adopted for MCM/MIW M&S).

Users and Uses of M&S

There are four categories of users for M&S in the defense establishment. These are:

1. Senior policymakers
2. Resource and acquisition managers
3. War fighters
4. Technologists and system developers

Each category has a different need and natural use for simulations. The *senior policymakers'*

M&S Definitions

Model: *A physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process.*

Simulation: *A method for implementing a model over time. Also, a technique for testing, analysis, or training in which real-world systems are used, or where real-world and conceptual systems are reproduced by a model.* (Note that there are several classes of simulation; typically one distinguishes constructive, virtual, and live simulations.)

constructive simulation: *digital computer model based*

virtual simulation: *simulation geared to immerse operators in rich visual audio/tactile environments*

live simulation: *simulation making use of actual equipment in a live-field setting*

Validation: *The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.*

Verification: *The process of determining that a model implementation accurately represents the developer's conceptual description and specifications.*

Accreditation: *The official certification that a model or simulation is acceptable for use for a specific purpose.*

principal benefits come from the ability of simulations to rapidly communicate difficult and complex concepts, whether they be system concepts, mission concepts, or concepts of operation. Depending on the skill and background of policymakers, constructive and virtual simulations can communicate in an hour or less very complex theater-level campaign plans, which might otherwise take weeks and even months of field time to execute. Simulations can rapidly, vividly, and audiovisually acquaint the observer with the minutest detail of own and threat weapons and systems, and their range of interactions. This is particularly relevant in the case of MCM, where few senior policymakers are exposed to what makes up an MCM operation and how MCM affects the larger theater of operation.

Resource and acquisition managers have more specific needs requiring simulation. The main natural uses here are:

- Force-level studies
- Operational Requirement Document (ORD) preparation and audit
- Cost and Effectiveness Analyses (COEAs), system design and other tradeoff analyses
- Rapid systems concept assessments and program prioritization
- Communication of study results and novel concepts to the senior policymakers

Principal and key users of simulation in the defense arena are the *war fighters*. War-fighters' concerns focus on training and tactical, operational and strategic planning, and concept development. To them, simulation offers a major adjunct to live exercises (live exercises being a form of simulation that is as close as one can get to war). In addition, different commands concern themselves with doctrinal issues. (In the Department of the Navy, the Naval Doctrine Command and the Marine Corps' Combat Development Command are the developers of doctrine.) In addition to the above, war fighters can use simulation as a means of operational plan evaluation and identification of planning and data gaps. Finally, the emerging advanced distributed simulation technologies associated with the Distributed Interactive Simulation (DIS) protocols utilizing the new IEEE Standard 1278 provide a means for tying together large numbers of participants in constructive and virtual simulations and live exercises over thousands of miles in highly coordinated training exercises.

The last category of users are technologists and system developers who use (higher fidelity) simulations to explore new technology and system concepts. The vision here is that, as more models are validated, and as more tactics and environmental data are captured in databases in digital format, virtual prototype construction would occur in times ranging from hours to days, shortening the development time and costs.

Advantages and Disadvantages of M&S

Authors of Reference 5 point out some general advantages of simulation as providing:

1. Increased accessibility to consequences of complicated nonlinear systems with many interacting parts
2. Possibility of discovering new phenomena by comparing experimental results to predictions of simulations; using simulation prediction as the basis for new experiments
3. Facilitating experiments that would be difficult or impossible to perform

However, they caution that sometimes attempts at extremely realistic models may end up defeating themselves by requiring very detailed measurements geared to satisfy a desire to incorporate as much "cellular" detail as possible. The danger they point to is that the more parameters and variables used, the more complex the representation becomes, ultimately making the model/simulation as difficult to understand as the real thing. They warn of the potential for invalid results arising from the inadvertent exclusion of important features (an associated issue here is that hyper-realistic models lead to high computational costs, either in central processing units (CPUs) or computational time). Based on these considerations, the use of simplifying models is sometimes suggested to enhance conceptual clarity and illustration of important principles.

In a similar vein, Reference 6 points out that if we consider hierarchical detail levels or strata, experience indicates that:

1. Models that relate variables of adjacent strata are the most powerful
2. Models that skip a level are difficult to test and are low in believability
3. Models that interrelate variables at a single level are weak in predictive power
4. Models that relate variables of several strata are most broadly significant

Additional Insights

Closer to military applications, John Hanley, a key individual in the naval wargaming arena offers the following insights:

GAMING:

1. Stimulates intellect, stimulates competitive instinct, provides occasion for social interaction
2. Used to teach and learn from
3. Used to study candidate courses of action in order to influence decisions
4. Allows meaningful, direct participation of top-level decision-makers
5. Provides rich insight into the structure of human behavior and decision-making
6. Exposes values and interests that conflict
7. Free-form gaming is particularly well suited to analysis of poorly understood, dynamic phenomena involving conflicting interests and human behavior

ANALYTIC MODELS:

1. Consist of a set of logical relationships
2. Permit manipulation according to formal logico-mathematical rules
3. Provide ease of manipulation, clarity of insight in the absence of firm data
4. Require structure, rather than reveal it
5. Try to capture only essential variables
6. Embody perspectives of the analyst/analyst-team
7. When applied beyond their limited domain, the models are often wrong

MACHINE SIMULATION:

1. Used when the analytic form becomes too complex, or where many calculations are needed
2. Embodies lack of transparency to users (may be there for programmers)
3. Requires effort resources to construct and populate databases

The Naval Mine Warfare Simulation Paradigm

Taking MIW considerations and the general experience developed in a variety of modeling and simulation efforts into account, an organizing picture or paradigm for modeling and simulation emerges. It is this paradigm that forms the basis for the approach taken for MIW simulation at NSWCDD/CSS (see Figure 8 for a summary).

Fundamentally, a landscape of *problems and issues* forms the context of discussions and effort. The *problems and issues* are translated into *needs and requirements*. Simulation is then a principal tool used to arrive at *realizable solutions*.

The term and domain of simulation are rather broad; consequently, there are many ways to dissect them. Part of the paradigm offered here is the division of simulation into two categories: *training simulations* and *analysis simulations*. Training simulations (tra-sims) are principally and immediately geared to development of personal skill of individuals (or linked groups of indi-

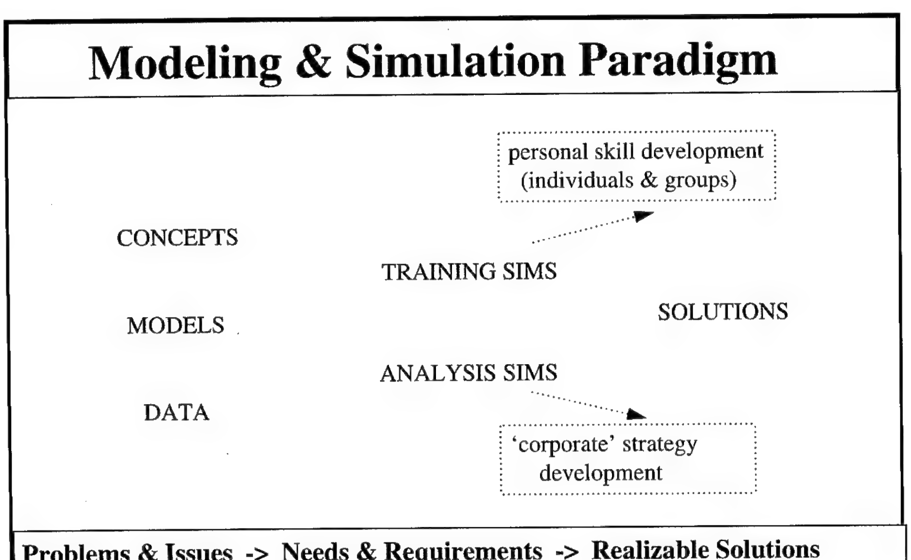


Figure 8. *Paradigm for modeling and simulation.*

viduals). Analysis simulations (a-sims) are geared to development of "corporate" strategy. (The term corporate refers to any organizational level from DoD and the U.S. Navy, through a component command, to a group of design engineers.)

Tra-sims are typically more tactical in nature and are oriented to having dynamic man-in-the-loop interactions, as well as man-in-control-of-hardware-in-the-loop. A-sims are more strategic in nature and are the ones that would be used for planning, engineering design, force-level studies, strategy selection, logistics planning, and COEA tradeoff analysis. Tra-sims typically run at real time (or slightly faster than real time) to accommodate human or hardware-in-the-loop response times. When connected to actual equipment-in-the-field, tra-sims must run at real time. In contrast, a-sims typically run much faster than real time. Tra-sims are often limited to relatively few, if any, replication runs, while a-sims require one or more order-of-magnitude more replication runs (typically to obtain robust statistical basis).

There are areas of overlap between tra-sims and a-sims. Either can (and should) be used as a source of informational input to the other. Both are motivated by the desire to obtain insights, experience, understanding, and solutions to pressing issues and problems. Both are characterized by dependence on input data, models, and concepts. The scope of the problem and solution requirement determine the level of resources (people, time, money, computers . . .) and detail that need to be applied. Tra-sims and a-sims should be regarded as complementary tools in the arsenal of simulation-based problem solving.

The naval MIW level training simulation development uses the Semi-Automated Forces (SAF) approach. This is patterned after the modular SAF (ModSAF) development and Synthetic Environment/Synthetic Forces experience obtained by the U.S. Army and the Synthetic Theater of War (STOW) programs managed by the Advanced Research Projects Agency (ARPA).

The MIW tra-sim development is part of a larger, multiservice countermine simulation development effort known as the Joint Countermine Operations Simulation (JCOS). The MIW a-sim is known as the Naval Mine Warfare Simulation (NMWS).

Joint Countermine Operations Simulation (JCOS)

The JCOS goal is to provide a capability to simulate countermine operations from deep water through an amphibious assault ending in army countermine operations. JCOS is a tra-sim designed to represent current fielded systems and highlight novel and developmental systems, particularly those U.S. Army, Marine Corps, and Navy Advanced Technology Demonstration Systems. (The reader is referred to CDR McBride of ONR for JCOS details.) It is also anticipated that this end-to-end countermine simulation capability will be prepared for use as a demonstration residual for planning and rehearsal purposes, and would allow experiential war-fighting concept and doctrine development. Currently, it is expected that major emphasis will be placed on integrating existing models and representations of the component systems, tactics, techniques, and procedures with: a) live instrumented systems, b) an after-action reporting system, c) 3-D visualization graphics, and d) Command, Control, Communication, Computers, and Intelligence (C^4I). A planned part of the JCOS development and integration is a robust Verification, Validation, and Accreditation (VV&A) effort. (The VV&A of the DIS aspects of

Originally conceived by the author, **JCOS** is managed by CDR Dennis McBride, Office of Naval Research, and is a pivotal element of the Joint Countermine Advanced Concept Technology Demonstration (CM-ACTD). JCOS development is executed jointly by the U.S. Navy, Marine Corps, and Army, and is integrated at MITRE using a team of Navy and Army engineers and contractors. JCOS development commenced in mid FY95.

NMWS is sponsored by the Program Executive Office - Mine Warfare (PEO-MIW) and is executed at NSWCDD/CSS. NMWS commenced in late FY94.

JCOS is a current Defense Modeling and Simulation Office (DMSO) project.) NSWCDD/CSS is the Naval Mine Warfare knowledge domain expert participant of JCOS development, as well as model supplier.

The Naval Mine Warfare Simulation (NMWS)

The Naval Mine Warfare Simulation is an a-sim category simulation. It is primarily oriented toward issues transcending the training of individual sailors. NMWS is principally oriented to provide assessment, policy-making, military planning, and acquisition support for the Naval Mine Warfare community.

The simulation is being realized as an extension of MARS that originated with the work of Henry Ng and Ken Wong of NSWCDD's White Oak Laboratory.[8] MARS software is coded in MODSIM II (Modular Simulation Language) (developed and Marketed by CACI Inc.), a high-level programming language. MODSIM is optimized for developing large process-based, discrete event simulations and makes significant use of object-oriented software technology. MODSIM is in some respects similar to Ada, Pascal, and Modula-2. MODSIM technology has specific benefits for module compilation and checking and for program control structures. A more extensive description of MODSIM II is provided in CACI's documentation.[9,10]

> Object-oriented programming is the current state of the art for large complex programs. Key elements include modularity and inheritance, which allow for *dynamic binding* at run time (i.e., dynamic selection of the code appropriate to the object). Another powerful aspect of MODSIM is the ability to provide for multiple inheritance "genealogies" for object construction.

The process-based, event-driven nature of the simulation allows for representing and accounting for many aspects of an object's behavior. Also, it allows creation of multiple concurrent instances of objects, operating simultaneously, with distinct parameter values. This is of significance to representing large number of mines individually. The unit of time used is dimensionless, allowing granularity ranging from nanoseconds to years. From a top level, object and entity change of states are the critical features, and since the simulation is event driven rather than time stepped, nonlinear time passage in the simulation becomes a key useful feature. This particular aspect is critical for running in a much faster than real-time mode, but still accounting for the arrow and passage of time. If needed, however, time-stepped subroutines can be incorporated into MODSIM, thus providing an extremely flexible software framework.

The MARS design philosophy accommodates both a man-in-the-loop (training/wargaming) mode and a noninteractive, or analysis, mode. The MIW component is particularly geared to emphasize analysis aspects. In the analysis mode, it is fundamentally a two-sided (although more sides can be assigned) Monte-Carlo, event-driven simulation. The MARS goal is to fully represent sufficient richness and detail of naval warfare in a multiwarfare, multimission environment. The original specification calls for an ability to include platform kinematics, weapon and countermeasure assignment and targeting, resource allocation, sensor representation and sensor and platform data fusion, battle damage assessment, command and control, and a variety of other functions too long to list exhaustively. These functions are part, at a high level, of MARS as an architecture pulling together:

- Data and component system/platform/environment models
- Pre- and post-processing for scenario-specific data input and output
- A simulation engine
- Graphical displays
- Distributed interactive simulation links

Within the Dahlgren Division, the White Oak and Dahlgren sites have developed MARS simulation capabilities and are extending functionality in the antiair, theater air defense, and surface warfare areas. CSS has implemented and is extending MIW and aspects of amphibious warfare functionality.

In the NMWS, the objective is to represent all aspects of the theater MIW and theater mine defense. This mode allows MCM Group

Commanders and Commander Amphibious Task Force (CATF) level integration and representation of events all the way to simulating interaction of individual mines with ships, sensors with mines, countermine systems and their supporting platforms with minefields, and obstacle interactions with a variety of platforms—all in the context of a dynamic environment.

In developing the MIW/amphibious extensions of MARS, the first step taken was to consider the MIW problem at a theater level to obtain a breadth of coverage. In other words, there was a focused effort to represent all platforms, systems, tactics, and operations that are relevant to the MIW problem. In effect, this covers all MIW-dedicated platforms and systems, as well as platforms that are susceptible to mines and obstacles. The next step of coverage is to bring representations of supporting elements and systems.

One of the key challenges in developing the simulation is the need to use models that are not trivial, but at the same time, not so detailed as to consume inordinate computer, time, and people resources. In modeling fidelity, one can consider a spectrum of levels of detail ranging from "back of the envelope or stubby pencil" to representing details at the quantum chromodynamic (QCD) level. Obviously the QCD level is not appropriate, nor is tracking every sonar ping or radar pulse at the total theater engagement level. To this end, the simulation development is pursuing a phased approach in several directions. The physical representation spectrum has been divided into three categories: a) parametric, b) standard, and c) improved. The first phase pursued development of a "parametric" fidelity level of description where object interactions are based on characteristic dimensions/lengths and characteristic probabilities.

For mine/ship/sweep interactions, the characteristic probabilities of actuation are obtained from running very detailed ship/mine or sweep/mine interaction models (such as the Total Mine Simulation System (TMSS)) and aggregating statistics over 10,000 individual ship/mine or sweep/mine encounters at different encounter speeds. The extreme detail acoustic, magnetic, Navier-Stokes, and other models are run "off-line" or independent of the main MARS simulation. Data tables are then generated and used as input.

There has been substantial discussion and debate over what the correct term and the correct level of representation should be. At CSS, we decided to adopt the three levels: 1. parametric, 2. standard, and 3. improved. Characteristic values for the parametric level are obtained from experimental observation or extensive off-line Monte-Carlo runs of individual high-fidelity physics models. Standard fidelity corresponds to standard sonar equation or radar equation level of detail. Improved fidelities go past the standard level and don't terminate at any specific resolution. Improved fidelities may range from molecular-dynamics simulation to in-situ hardware-in-the-loop.

At the parametric fidelity level, care is taken to make sure descriptions of all levels are consistent. In other words, while individual platform kinematics can be described in great detail, this is not critical nor advantageous; thus, kinematics are described by way-point progressions with attendant navigational drift. Associated with systems and platforms are tactics; these, too, get represented. Here again, care is exercised to obtain a meaningful, but not overwhelming, level of detail.

The next, or standard, fidelity level uses representation for interactions similar to performing design calculation with a sonar equation, with individual terms calculated using environmentally dependent parameters. Thus, one would start with frequency-dependent source levels, directivities, target strengths, etc., and calculate propagation losses, reverberation, noise levels, returned signal levels, signal-to-noise ratios, etc., and pass those to detection models.

The final level of representation in our terminology is "improved fidelities." This level captures representations beyond energy equations (such as sonar equations), and allows finer detail description of interactions. For example, in optical modeling, one would follow the life history of individual photons from emission, propagation, and capture by sensors, through conversion to image pixel maps. The processing models would then incorporate entire processing algorithms through detection classification and identification

modeling. If required, this level of improved fidelities would include sensor and weapon components or even entire systems in the loop coupled to human operators in the loop for system/sensor decision-making.

The main categories of models required for a comprehensive description of MIW are:

1. Physical and environmental phenomenology
2. Platform-related models
3. Sensor models
4. Weapons and countermeasure models
5. Command and control models

A partial listing of the coverage of individual models required for MIW is given in Figure 9. Worth noting is that, while physical equipment and physical phenomena are quite challenging to describe, by far the most complex elements to represent are the contingent command and control relationships and interactions. These must embed tactical, operational, and strategic concept and doctrine aspects that are both documented and experimental. In this context, it is also important to recall the distinction between commanders [who develop and adjust strategies] and operators [who maneuver ships, fly planes, and man the tactical displays and consoles]. (This distinction is articulated well in Reference 7 and is key in considering who you invite to wargames, simulation exercises, and fleet exercises.)

Obviously, the program sketched above is quite demanding and will take time to bring to completion. At this time, a significant portion of acquisition-related MCM elements have been captured in the parametric level of representation. This parametric level is known as MARS(D) Version 2.0. Version 2.0 has been exercised in a major MCM Simulation Exercise (SIMEX 95-1) oriented at exploring issues associated with preamphibious assault MCM operations and early stages of the ship-to-shore movement of the amphibious assault.

SIMEX 95-1, conducted during spring 1995 at CSS was the first major MCM simulation, and involved participation by Fleet commanders and planners. Red and Blue Cells were chartered by Commander, MIW Command. These cells prepared their respective plans and laydowns. CSS analysts and simulation staff implemented the Red and Blue actions. Extensive simulation data were collected and analyzed. Results are being prepared for formal briefings and reports at this time.

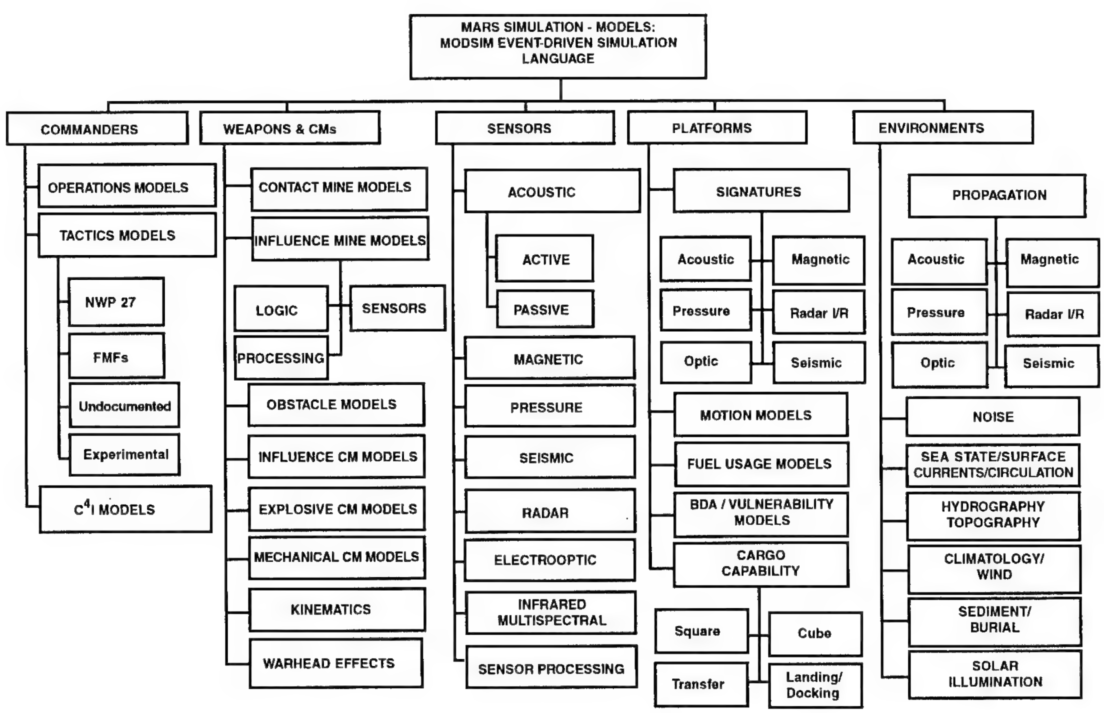


Figure 9. *Models for MCM simulation.*

In conducting the exercise, several subscenarios were assessed including conducting:

- Amphibious assault in the absence of mine threat
- Conducting preassault MCM operations in the absence of mine threat
- Conducting the amphibious assault (with threat mines present) but without preassault MCM operations
- Conducting MCM operations followed by the assault

The diversity of the subscenarios then allowed interesting comparisons to be made to assess the contribution of MCM (when needed) and the penalty of conducting MCM operations (when they are not needed). A number of interesting lessons were learned by all parties involved in constructing the SIMEX (i.e., the war fighters, technologists, engineers, and analysts). A key gain for the MIW community was creating meaningful reference scenarios with a sufficient level of detail for future studies and wargames to use. Finally, the baseline subscenarios form a foundation for a variety of excursions involving consideration of alternatives such as different levels of reconnaissance operations or substitution of alternative (novel) systems.

One is now able to pose and answer a variety of questions involving quantitative and functional measures such as:

***QUANTITATIVE MEASURES*:**

- Rate of advance
- Ground position
- Status of support forces
- Status of sustainability
- Status of control structure
- Attrition rates:
  - key weapons/weapons systems
  - forces
- Casualties

***MEASURES OF FUNCTIONAL CAPABILITY*:**

- Prospects for completing ongoing offensive/ mission
- Prospects for defeating enemy's plan
- Capability to intercept a large fraction of attacker's force and limit damage
- Capability to deny enemy use of particular regions

Following the experience with SIMEX 95-1, the Fleet requested consideration of additional scenarios; also, several research and development (R&D) projects have now started to use MARS(D) for investigating revolutionary advanced concept applications for MCM.

Version 2.0, as well as the SVP visualization tools described below, are being provided to the Naval Decision-Making Support Center at the Naval War College as part of CSS' total system engineering support of the War College's advanced M&S facilities. As an informational point, Version 2.0 has been ported to a number of Unix workstations (DEC Alphas, SGI, TAC-3).

Visualization and Fiske's Challenge[11]

Many years ago, Admiral Fiske remarked that:

No man ever lived who could describe a complicated machine accurately to a listener, unless the machine differed but little from a machine with which the listener was acquainted. But hand a drawing of even a complicated machine to a man who knows its language—and the whole nature of the object is laid bare to him.... So, when the forces representing a complicated Naval situation are placed upon the board-game, all the elements of the problem appear clearly and correctly to each person; the imagination has little work to do, and the chance for misunderstanding is almost negligible.

Fiske recommended that a detailed plan for every contingency should be prepared where:

each distinctive approved solution would be photographed in as small a space as practicable, preferably on a moving picture screen . . . . These photographs, shown in appropriate succession, would furnish information analogous to the information imparted to a chess student by the statement of successive moves in those games of chess one sometimes sees in books on chess and in the newspapers.

Fiske's visionary challenge is now a reality. A unique visualization tool called Simulation Visualization Program (SVP) was developed at CSS that allows much of what Fiske called for.

SVP version 3.0 is a unique computer program (running on SGI-ONYX) designed for visualization of air, land, and sea aspects of warfare with a dynamic ocean/wave visualization

model permitting controlled variation of sea-states. The program provides DIS compatibility, functionality, and connectivity through interpretation of standardized Protocol Data Units (PDUs). SVP allows inclusion of a variety of terrain and underwater topographies, individual control of any number of objects (depending on computer memory limitations), including terrain objects such as plants, bushes, trees, etc. Extensive naval warfare elements are of particular interest to the MIW community (mines, obstacles, and MCM equipment). It allows preloading of objects as static or dynamic objects with trajectory history tracking for dynamic objects, as well as recording and automatic playback of scenarios. Controllable visualization of waves with respect to sea-state, opacity, and wavefront components are also available. Multilevel representation of objects allow conservation of computing resources as a function of perceived nearness or remoteness of objects.

SVP also contains a large set of graphic user interface controllable perspective points, fields of view, lighting, etc. In development, are advanced satellite imagery to topography clamping routines, as well as enhanced wave and ocean interaction representations (foams, whitecaps, waves, and wakes generated by seagoing platforms) and interaction of waves with vessels.

An extensive library of air, sea, and land military platform icons has been assembled and is continuing to be developed (where possible, icons are purchased from commercial off-the-shelf sources).

The SVP program was tested for connectivity to MARS(D) in January 1995. Figure 10 shows the SVP realization of the COBRA Advanced Technology Demonstration (ATD) system; Figure 11 shows an early stage of SVP depicting LCAC operations. Currently, we are gearing up for full connectivity with all aspects of MARS(D)-V2.0 for use in visualizing the simulation exercises conducted and in process. With this step complete, the Navy is well on its way to meeting Fiske's Challenge.

Issues and Challenges

In looking both forward and back at some of our efforts, it is clear that many issues and challenges remain. One quickly realizes that the task of representing warfare is enormous. In the past, available technology posed a severe limit on how much could be represented and coupled at any one time. Accelerating computer capabilities and declining computer costs now move this limit to where technology is a friend (and only cost is a hindrance). All recognize the value and utility of simulation and the savings to be realized, in comparison with extensive field exercises. However, representing human interactions and human decision processes will always pose a challenge. In the same vein, representing human mental states, such as morale or deterrence, will need increased R&D attention.

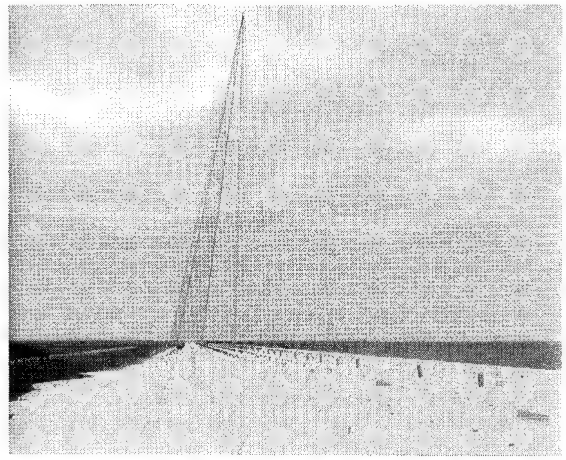


Figure 10. *Simulation Visualization Program (SVP) representation of the Coastal Battlefield Reconnaissance and Analysis (COBRA) ATD prototype. COBRA is designed to detect minefields using electro-optic technologies.*

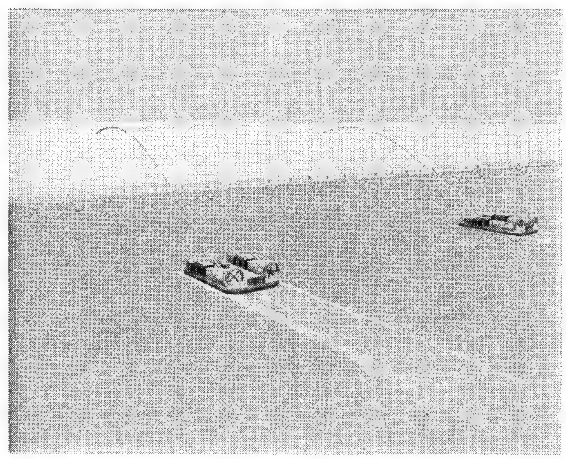


Figure 11. *SVP representation of LCACs configured as MCACs deploying line charges to clear surf and beach zone mines and obstacles.*

Developing rigorous methodologies for using large simulations will require new paradigms and new skills. Very few people today can comprehend extensive, detailed descriptions of any one warfare area, let alone several distinctly unrelated warfare areas working together. Training a cadre of "big-picture, high-detail" analysts is key, as well as certifying users of large simulations (after all, the work of these folks will be pivotal for those who make billion-dollar decisions).

There are many software and hardware technology-oriented issues; these include developing a strategy for use/reuse and portability of software across a number of computing platforms (and new generations of computer hardware). Also key is improving high bandwidth connectivity for networked simulations to allow much faster than real-time play of simulations.

Finally, a shared vision is required to bring the theoretical and practical elements of warfare into the twenty-first century and assure its success in the face of what might be termed strategic uncertainty. This is especially relevant to the emerging domain of expeditionary warfare, a domain currently being formulated at the Office of the Chief of Naval Operations. Expeditionary warfare is the type of warfare that the Navy and Marine Corps must execute in the future when dealing with remote, land-based threats as part of larger Joint Littoral Warfare/Joint Expeditionary Warfare settings. I would like to recommend as a possible approach to finding the shared vision what I call the "Wargaming-Simulation-Fleet Exercise Continuum." In this continuum, technologists, engineers, acquisition managers, and war fighters work in a phased, but repeating, spiral of wargaming to define detailed simulation exercises that, in turn, motivate specific fleet exercises which, in turn, feed inputs and concepts to wargames, and data to simulation models.

Conclusions

A systematic effort to simulate the breadth of MIW operations was described. Elements of progress and accomplishments with MARS were presented. MARS(D)-Version 2.0 was completed and used for the conduct of Simulation Exercise 95-1. Related efforts associated with MIW simulation, including an advanced simulation visualization program currently being coupled with MARS, were also described. Finally, challenges, issues, and vision for the use of modeling and simulation in an expeditionary warfare context were offered.

Acknowledgments

The efforts described here have been motivated by encouragement and sponsorship of PEO-Mine Warfare (especially Dr. Dale Uhler, then Deputy PEO-MIW, now Deputy Commander, Space and Naval Warfare Command, and Mr. Charles Hill); RADM John Pearson, Commander, Mine Warfare Command; Dr. Frank Shoup, Associate Director, Expeditionary Warfare Division (OPNAV); Col. Joe Singleton and Mr. Robert Feden, Office of Naval Research (ONR 32), as well as NSWCDD/CSS' senior management. Program sponsorship originated at NSWCDD/CSS with in-house support for the SVP program and initial phases of the MIW simulation design; the principal programmatic support is provided by PEO-MIW; and support for JCOS and some aspects of MIW modeling is provided by ONR. The technical progress described here would not occur without the initial assistance of NSWCDD/White Oak Laboratory in providing the original MARS software to CSS, and the continuing collaborative spirit and dedication of a large team of modeling and simulation specialists at the Panama City and Dahlgren sites of the Dahlgren Division.

References

1. Hartmann, K. H. and Truver, S. C., *Weapons that Wait; Mine Warfare in the U.S. Navy*, updated edition, Naval Institute Press, Annapolis, 1991.
2. Shepherd, D., *MCM Functional Breakdown,* MCOG, NSWCDD/CSS, 1995.
3. Buckley, T., and Shepherd, D., *Theater Mine Defense,* MCOG, NSWCDD/CSS, 1995.
4. Slatkin, N., memorandum for distribution, "Policy for Modeling and Simulation," Assistant Secretary of the Navy (Research, Development and Acquisition), 3 Jan 1995.
5. Sejnowski, T. J.; Koch, C.; and Churchland, P.S., *Science,* Vol. 241, pp. 1299-1306, Sep 1988.
6. MacGregor, R. J., *Neural and Brain Modeling*, Academic Press, 1987.
7. Hanley, J., *ON WARGAMING,* Ph.D. Dissertation, Yale University, 1991.
8. Ng, H. C., *Multi-Warfare Assessment and Research System (MARS) Requirement Specification WSA&E Low Fidelity, Version 1.5,* NSWC TR 91-746 (DRAFT), 1 Aug 1992.
9. *MODSIM II The Language for Object-Oriented Programming Language: Reference Manual,* CACI Products Co. La Jolla, CA, 1994.
10. *MODSIM II The Language for Object-Oriented Programming Language: Tutorial,* CACI Products Co. La Jolla, CA, 1994.
11. Fiske, RADM B. A., *The Navy as a Fighting Machine*, Charles Scriber's & Sons, NY, 1916.

The Author

ELAN MORITZ, research physicist, graduated with a B.S. in physics from State University of New York at Stony Brook, and with a Ph.D. in physics from Kent State University in Ohio. Prior to joining NSWC/CSS in 1981, he was a senior engineer at EBASCO Services in New York City. As Lead Applied Physics Engineer there, he contributed to the design of the Tokomak Fusion Test Reactor (TFTR) at Princeton University and a number of commercial nuclear power plant design and safety analysis evaluations. At CSS, he headed a joint multinational development and MCM demonstration program that resulted in fielded hardware and the current international standard for detailed mine/ship/sweep encounter simulation. Later, for over five years, he directed the Navy's Exploratory Development Program for surface and submarine sonar and torpedo countermeasures. This program resulted in a number of systems and system improvements now in the Fleet. Following that, he headed the Signal and Image Processing Branch at the laboratory, leading to significant CSS capability in computer aided/automated detection and classification especially in electro-optic and sonar image understanding and processing. Dr. Moritz has served as Technical Director-Mine Warfare (OP36T and N852T). His awards include the TTCP Achievement Award and the Department of the Navy's Meritorious Civilian Service Award. He is the first Director of the Coastal Systems Station's Modeling and Simulation Office (CMSO). In this capacity, Dr. Moritz is responsible for formulating policy and executing CSS' corporate vision of modeling and simulation. He also advises various senior Navy offices in areas associated with mine warfare.

Vision of an Integrated Systems Development Approach

Robert E. Podlesny

Today's world is dramatically different from the bipolar world in which the current defense establishment was created. Reduced funding and the resultant shrinking of the uniformed services, the civilian support structure, and the industrial base are today's norms. Yet, this shrinking has not eliminated the need for the defense establishment to be prepared to defend U.S. interests in a multipolar world armed with "high-tech" systems, many of which are weapons of mass destruction. A capability is needed that can address such fundamental issues as: the rapid use of emerging technologies; innovative, team-structured management; and rapid prototyping methodologies in order to deliver quality products to the Fleet in a streamlined, efficient manner. This capability encompasses ***all*** *of the challenges facing the surface Navy, including the traditional warfare areas, as well as the dimensions of information and time (Figure 1).*

The purpose here is to describe a holistic engineering concept for addressing the development of complex systems. This ***notional*** *approach includes all key players, operators, developers, and acquisition managers. As warfare in general, and naval warfare in particular, becomes more complicated, so does the need for rapidly exploiting technology. This article attempts to describe an environment that supports the efficient acquisition of naval systems and the rapid exploitation of new technology.*

Background

The *Defense News* reported in its July 31, 1995 edition that a study prepared for the Chief of Naval Operations, Admiral Boorda, found there is "no systematic process in the Navy for thinking about major innovations in naval warfare or for speeding their introduction to the fleet." This article attempts to describe an environment that supports the efficient acquisition of naval systems and the rapid exploitation of new technology. This concept is also synergistic with the "Joint Modeling and Simulation Evolutionary Overview," which was prepared by the Deputy Director for Technical Operations Force Structure, Resources and Assessment Division, and signed by General Shalikashvili in February 1994. The approach has been tentatively called the Technology Infusion/Complex Systems Engineering Environment (TI/CSEE).

The Challenge

As noted in the 1994 Naval Resource Advisory Committee report to the Assistant Secretary of the Navy (Research, Development, and Acquisition (ASN(RD&A)), the Department of the Navy must "revolutionize" the way it supports the development and acquisition of weapon systems. The ponderous bureaucracy that identified a *specific* strategic or tactical need; developed reams of supporting requirements documentation, specifications, requests for proposals, and contracts; conducted an endless cycle of reviews; and took years to prototype, test, and retest is going away. The old, structured *linear*

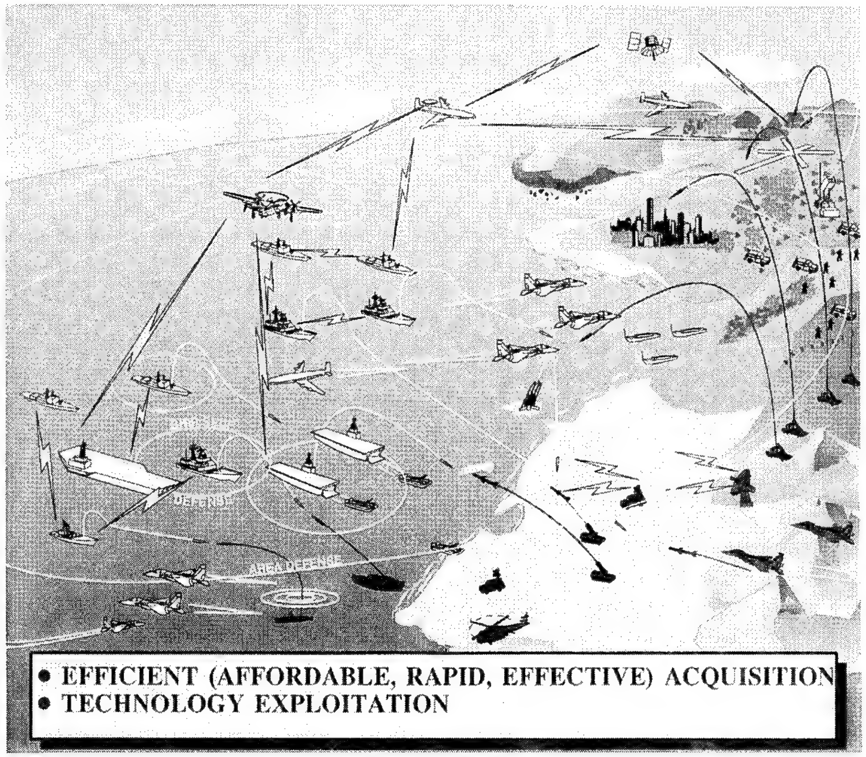


Figure 1. *The Technology Infusion/Complex Systems Engineering Environment (TI/CSEE).*

approach to problem solving is now *inadequate*. Today's weapons must be developed faster, *perform better*, and interact jointly, within a complex system of systems.

The Approach

As Alvin and Heidi Toffler have noted in their book entitled, *War and Anti-War, Survival in the 21st Century,* developments in military technology track the development of a country's economic state. As society moves toward a greater distributed, information-based economy, so must the U.S. military. The weapons development focus must shift from how a given sensor, weapon, or processor, etc., works in a "stand-alone" environment, to the entire *integrated* environment. This change in thinking is required to address the complexity of modern society and warfare. Systems requirements must be addressed *at all levels* of the development cycle. The piecemeal approach that designs and acquires weapons to counter a specific threat must be abandoned. The ability to address constantly changing threats and requirements, *and* rapidly evolving technologies is needed. A lean, *iterative* acquisition methodology that supports the insertion of new or improved components into weapon systems at various stages in the systems' life cycles is needed. This environment must also contain the ability to prototype and assess the flow of information *throughout entire* systems; e.g., strategic and tactical. Victory on tomorrow's battlefield depends upon the successful synthesizing of *key* data within the "time-critical envelope." The methodology itself must take advantage of the technology we have today. Such an environment can also help conceptualize the scales and complexities confronting the way the system may be employed in potential conflicts.

To exploit new technology, the design, prototyping, and testing of new systems will have to be performed *concurrently*. As the ASN(RD&A) noted in a recent vision statement, modeling and simulation will play a key role in the acquisition revolution. Complex systems engineering efforts should pursue a *dual* model-test-model approach. This dual approach involves *two mutually supporting paths*: one, a *technology exploitation/rapid prototyping* path; and the second, an *operational* path based on *existing* systems. A comprehensive toolkit consisting of: (a) operational/tactical equipment,

(b) constructive models and simulations, (c) virtual representations, and (d) advanced distributed networking technologies, is needed (Figure 2). The *complete toolkit* needed to tackle today's and tomorrow's environments must also embody a key philosophical element—"Complex Systems Thinking."

Traditional parochialism *cannot* be tolerated in today's acquisition world. The *entire* operational arena must be conceptualized when addressing the role that a new or updated weapon system will play in support of the Navy's overall mission. This includes not only the *technical performance* of the hardware and software, but other key factors such as: *operational suitability*; *tactics* and *doctrine*; and *interoperability* within the host ship, battle group, and joint theater. These considerations must be included in the design and test process *from day one*. Affordable high-performance processors, innovative object-oriented, open-architecture software designs, and universal data bases are revolutionizing modeling and simulation capabilities. These tools are allowing high-fidelity models to be used not only by developers, but also by the training, test, and operational communities. Developments such as these are also providing the basis for the inclusion of the key element of virtual systems: *reality*. The ability to insert *realism* at all stages of the acquisition and development process is rapidly approaching (Figure 3).

The Road Map

The implications of comprehensive systems thinking can be daunting if tackled in one fell swoop. The intricate relationships between subsystem managers and developers is itself tangled enough to thwart this concept. The following is a stepwise road map that lays out an incremental development process.

Creation of a Technology Infusion/Complex Systems Engineering Environment

The design and procurement of weapon systems is too important to be left to any one technical community (e.g., engineers, accountants, warfighters, lawyers, et al.). Complex system challenges require holistic thinking by an *integrated* team of experts, armed with affordable, relevant, and believable tools. The TI/CSEE approach integrates a network of distributed technology and tactical "cells" consisting of representatives from key organizations such as: the numbered fleets, training, and doctrine personnel; System Command (SYSCOM) program managers; research and development (R&D) and engineering facilities; and contractors.

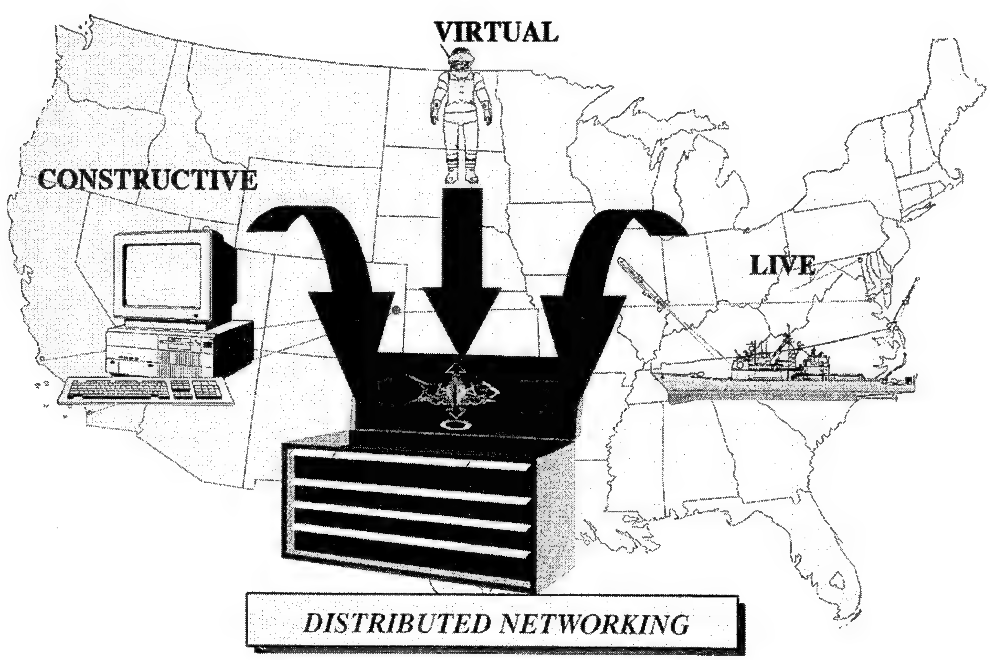


Figure 2. *Toolkit and infrastruture.*

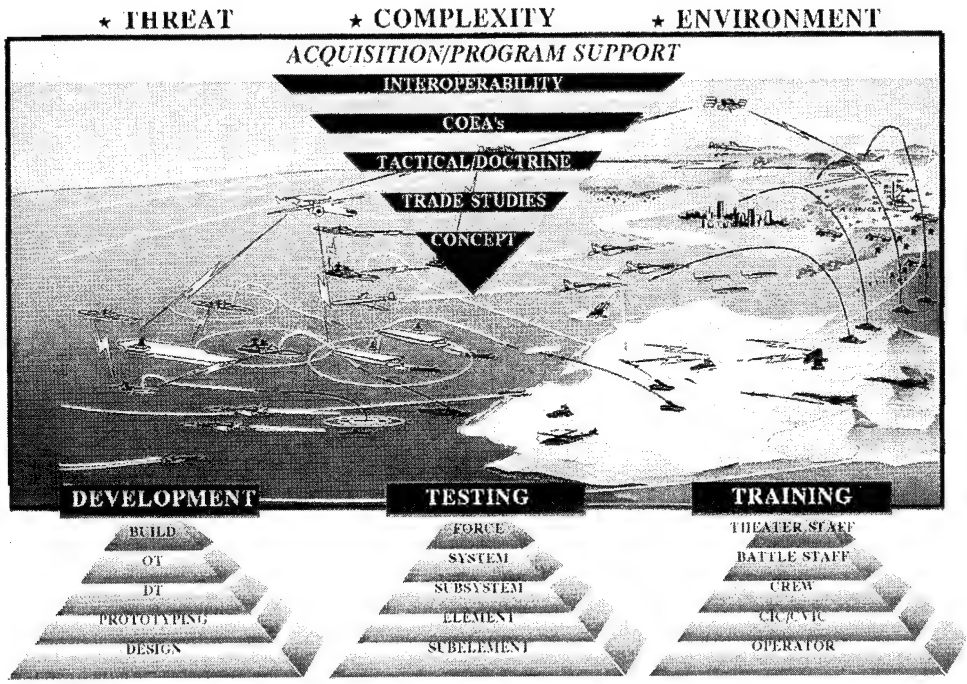


Figure 3. *DoD modeling and simulation master plan.*

The use of this type of environment will help visualize concepts and expand the horizons of the development team by providing them with prototype/hybrid systems, not just "vu-graphs." As noted, a dual-path approach is recommended to start the realization of the TI/CSEE.

Operational Path. Start with "today's" baseline system(s). Seamlessly connect existing physical assets (e.g., laboratories and shipboard equipment) in a distributed hardware-in-the-loop (HWIL) network. The "creation" of tactical elements such as individual ships or even entire battle groups via the networking of distributed, shore-based assets, advanced simulation resources and prototyping assets. The integration of existing tools such as the Battle Force Tactical Trainer (BFTT) with systems engineering efforts is one example of an ongoing effort. This type of integration is needed to address today's Navy and Joint concerns especially in the Command, Control, Communications, Computing, and Intelligence (C^4I) arena. This networking of tactical systems can be a cost effective way to address: limited ship and personnel availability; providing Joint architectures; and injecting threats and environmental factors into systems engineering studies, tactics and doctrine analyses, training scenarios, and test and evaluation exercises. The integrated development community should have access to this net.

Technology/Rapid Prototyping Path. Upfront systems engineering must be more than a collection of paper studies. The process must be evolutionary and be able to integrate tactical requirements with technology needs and capabilities, and produce a useful tool that actually *infuses technology* into the tactical world. Once potentially useful technologies have been identified via an enhanced Basic Technology Review effort, they should be modeled/simulated and or prototyped, then assessed via the use of the TI/CSEE. This path will allow for the rapid and concurrent development of conceptual and prototype components and systems. R&D facilities and contractors will be able to link with this tool to "integrate" their products with other new or existing capabilities. This may be viewed as a *developmental test bed or platform.* Requests for Proposals may actually be illustrated/ demonstrated by the Government for prospective bidders, who, in turn, may actually "fly" their systems via models and simulations or prototypes for evaluation during the proposal evaluation process. The participation of operational personnel is particularly needed at this stage of the acquisition process in order to streamline the

development of new systems by helping to assure the requirements are still valid and are being adequately addressed. Once a concept/ prototype is accepted, it may be "unplugged," and replaced with a "plug-in" of the "real" product.

Creation of a Networking Infrastructure

The linking of today's naval weapon systems via wide, area networking technology is mandatory. Networked land-based facilities (i.e., laboratories, wind tunnels, training centers, model basins, anechoic chambers, test ranges, etc.) can be used to supplement operational training and testing requirements. Networking sea-based assets (i.e., ships, submarines, airplanes, etc.) can, in turn, be used to add realism and the "operator's touch" to R&D activities. The sharing of resources will save redundant costs and help alleviate the workload on overtaxed systems and minimize the impact to any one program. Meaningful and frequent use of these assets will also help drive down the operational and maintenance cost of these networks. The network infrastructure will be linked by dedicated or dial-up communications lines (i.e., copper, fiber, wireless, etc.) and/or existing networks such as the Defense Simulation Internet (DSI). These distributed resources will communicate via accepted network software (i.e., Aggregate Level Simulation Protocol (ALSP), Distributed Interactive Simulation (DIS), Higher Level Architecture (HLA), etc.). The bottom line is that these networks of naval assets must be *sensibly employed* in order to streamline the acquisition system and realize cost savings. To be truly effective, the Navy's acquisition process must take advantage of *commercial practices* and capabilities.

Establishment of a Universal Modeling and Simulation Data Base

The core of this TI/CSEE will be data bases of constructive and virtual models and simulations. These models and simulations range from parametric representations to high-fidelity engineering models. They depict environmental factors; sensor physics; weapon performance and kinematics; command, control, and communications links; threat and friendly forces; approved Department of Defense (DoD) crisis response scenarios; and other capabilities. The effort will rely on existing "cells" of expertise located at military organizations such as the Training and Doctrine Command (TRADOC), government and university labs, and at contractor facilities to develop and house these models and simulations (Figure 4). This distributed "library" may be utilized by any approved cell or node with access to the network. The creation of this library of databases, and models and simulations, should be initiated as the first step toward a master "Technology Bulletin Board," which will be available to support both simulation-based design, and acquisition activities.

Integration of Elements with the Overall System

This backbone of data bases, tactical equipment, and other appropriate resources will complete the toolkit from which acquisition management will draw needed resources. System-level program managers will rely on subsystem and element developers to actually produce the components of their systems. The overall system will be built and integrated using a distributed architecture. Control and validation of subelement models will remain with the developing agencies. The integration of the system will be the responsibility of the system program manager. System requirements will be addressed by the appropriate program manager.

Each layer of this toolkit will rely upon its own set of analyses and models and simulations. Design level tools will consist of the highest fidelity models, while the tools used to "fight" a theater war game will consist of lower fidelity, broad spectrum models. The layers in between will be tailored to meet their own requirements. This layering or hierarchy is described in greater detail in Figure 3.

With the continued rapid growth in computing technology, there may be a mixing of the capabilities across the levels described above to the point where the entire development process is supported by the same tools.

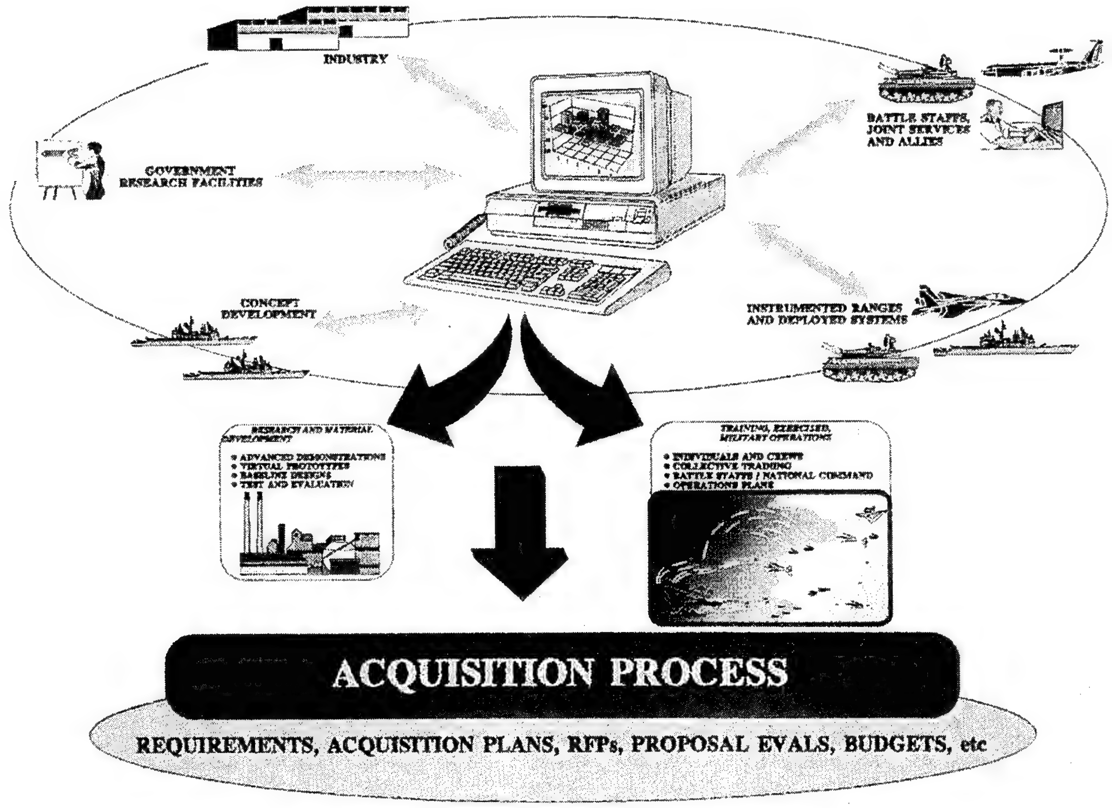


Figure 4. *Universal Database.*

Verification, Validation, and Accreditation (VV&A)

As noted, subelement program managers are responsible for the VV&A of models and simulations that represent their components in the distributed library. An overall VV&A plan that identifies how models and simulations will meet the "affordable, relevant and believable" is required. This plan must address the fact that models reside in multiple locations, and were probably developed for varying reasons and to varying standards. The overall plan must assure the acquisition and operational communities that, even though their development and exercise efforts may be complex and distributed, they are being executed on a "level playing field" (Figure 5).

Another key aspect to be included in this effort should be the use of a standard set of processes. These processes will address tool development standards and help ensure that the desired level of commonality is achieved across the entire network. The development of these processes should be completed by a group of representatives from each of the major nodes on the network. The various organizations should, in turn, develop their own in-house processes. This task is analogous to the "Software Maturity Capability Model" developed by the Software Engineering Institute at Carnegie Mellon University.

The Uses

Concepts of Operations Development

The Concept Evaluation Phase will use the TI/CSEE to develop Concepts of Operations (CONOPS) documentation and a notional prototype. The use of automated and visual tools will be maximized. Operational requirements will be illustrated via the use of modeling and simulation tools. User interfaces and parametric analysis will be available for programmatic and operational decision-makers. Given weapons and systems may be modeled and simulated, and their applications evaluated in support of tactics and doctrine development. The culmination of this phase will be the decision to continue or

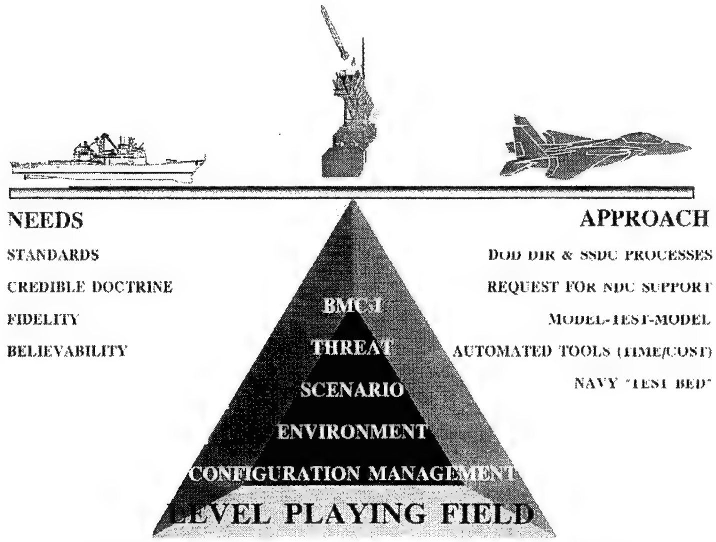


Figure 5. *Verification, validation, and accreditation.*

terminate the development of the system in question. As discussed earlier, traditional operational evaluation (OPEVAL) is basically a test to determine if a given system can be operated in the Fleet by uniformed personnel. This test has traditionally been conducted near the very end of the development cycle, *after* millions of dollars have been spent. This is too late and inefficient. If the concept and the ability of the system in question cannot meet the requirements of operational forces, it should not be built. If the system being developed meets the requirements and can notionally complete its mission, then further development will continue. If not, cancel or modify the effort.

The TI/CSEE may be used during this phase as the prototyping environment. As an example, the TI/CSEE could host a prototype combat information center (CIC). This laboratory CIC may consist of reconfigurable workstations loaded with prototype graphical user interfaces that represent tactical hardware and software. Off-board sensing and C^{4}I data may be provided by local or remote sources. Operational plans and scenarios will be ported in from TRADOC or the Operational Test and Evaluation Force (OPTEVFOR), etc. Warfighters will be able to "see and feel" how a notional system will work in a "real" environment *prior* to the commitment of further funding (Figure 6).

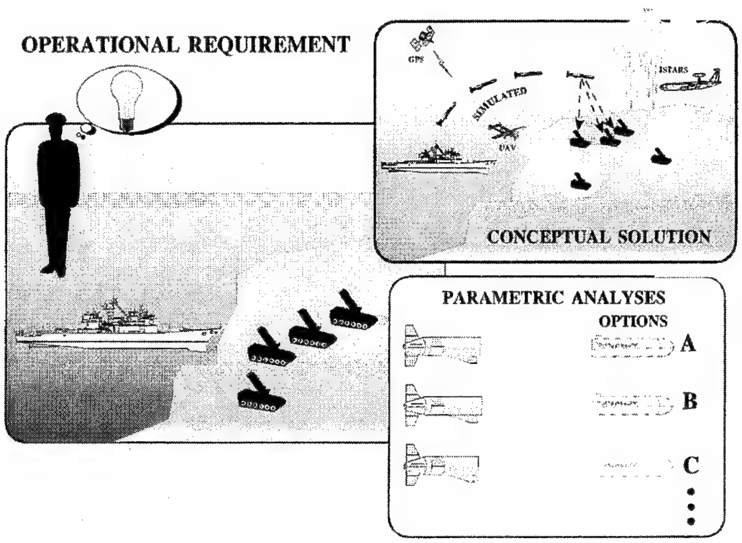


Figure 6. *Concept evaluation phase.*

Systems Integration

As the concept is proven and the operational suitability determined, further systems integration will continue in earnest. The integration of HWIL, prototypes, C^4I systems, and the like, should be incorporated here. This integration should be scenario driven and include not only U.S. forces' connectivity, but should also address potential threat laydowns and environmental factors. Actual bench-level developers must work closely as part of an integrated team with operational end users, program staff, and contractors (Figure 7). The technical performance must constantly be measured against tactical utility. Paraphrasing Soviet Marshal Zukhov: *design and build the weapon the way you will fight with it.* It is much cheaper to reengineer a system at this stage than further along in the development cycle. If a given technical approach cannot meet the operational or cost restrictions, then a reassessment by the entire development team needs to occur.

Test and Evaluation (T&E)

The TI/CSEE will provide a capability to support realistic system-wide assessment anywhere in the world in any environment against any threat in any scenario—a true operational test. *Final technical performance* testing will still need to be conducted in the field with actual gear to assure that delivered products do, in fact, behave as designed. Results from the system integration phase may be used to augment this effort, as will links with live systems and techniques such as "captive carry" events and Joint operations. The "Synthetic Test Range" program is one example of how this is being addressed today (Figure 8). The use of simulations and models to stimulate systems is growing. The Joint Warfighting Center (JWFC) is developing the Joint Theater-Level Simulation (JTLS) tool to stimulate the Global Command and Control System (GCCS) to enable Joint service operators to conduct operational training and planning exercises on their tactical equipment. Just as the integration effort is scenario driven, so should the testing and training efforts.

Training/Fleet/Life-Cycle Support

Overall life-cycle support material and processes should be derived and supported by the TI/CSEE. Documentation, maintenance and repair activities, and training are just three examples. Operator and technical manuals should be provided in automated formats derived from the TI/CSEE development process. Where appropriate, this data should be in multimedia formats. What better way for a sailor to learn how to perform organizational maintenance than by watching it being done on a CD or other video media? Some Navy programs are already using CD technical manuals. The use of holograms would also be useful to support this effort. A three-dimensional replication of hard-to-reach machinery spaces or cable runs are just two examples of where holograms could be used (Figure 9).

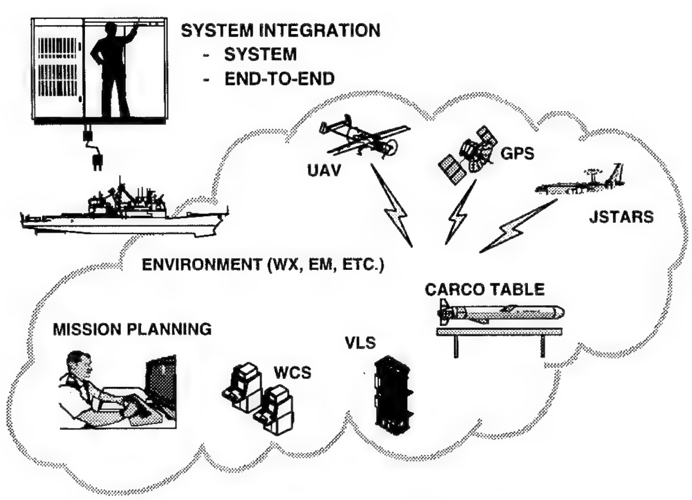


Figure 7. *System intregration phase.*

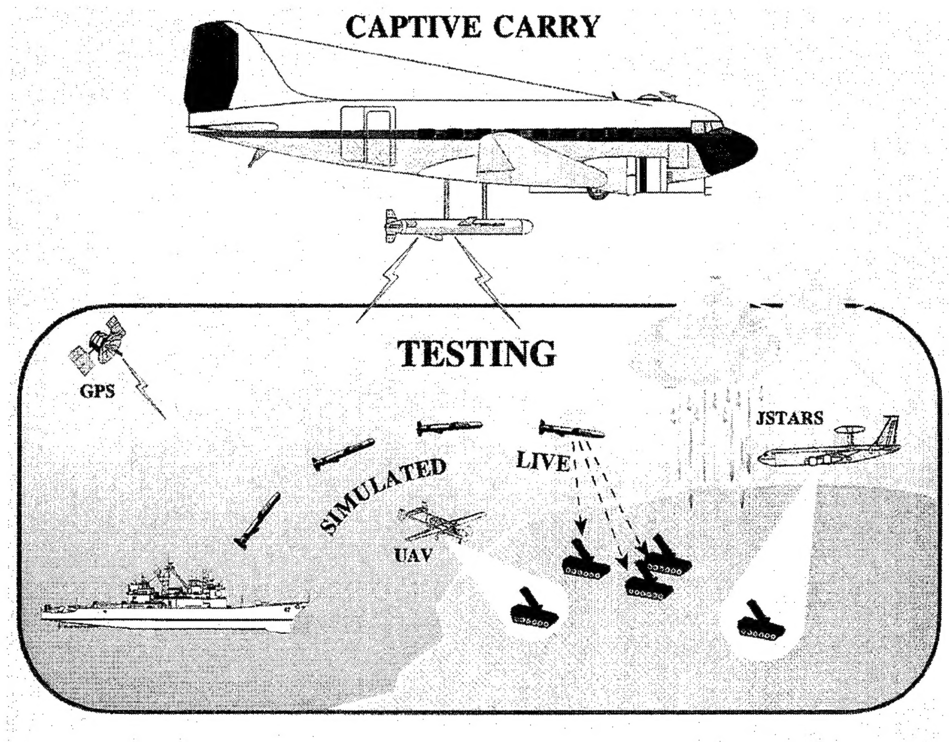


Figure 8. *Test environment.*

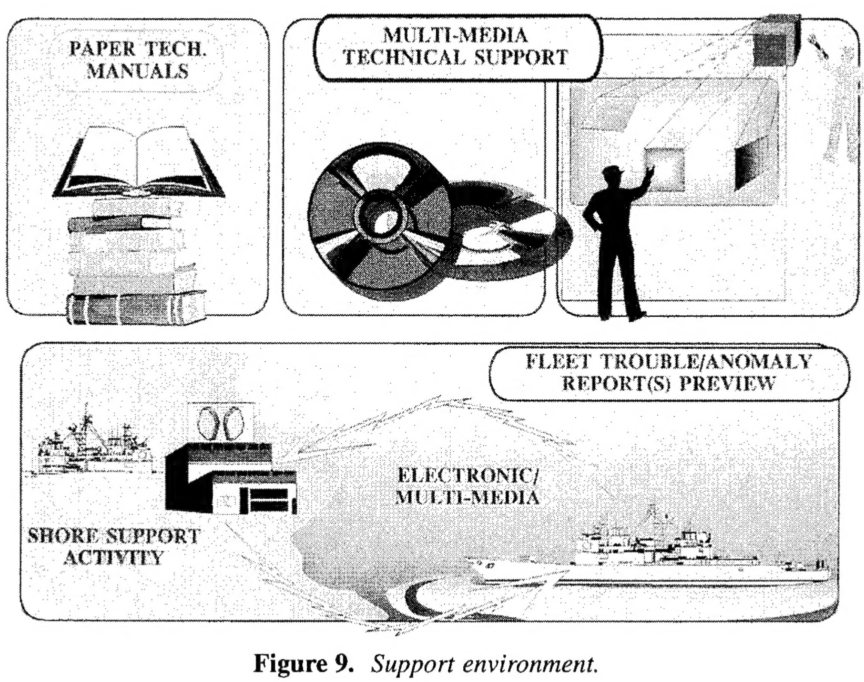


Figure 9. *Support environment.*

Systems should be *designed* to facilitate remote anomaly troubleshooting. The space program has been doing this for 30 years across vast distances. This capability will allow the seaborne members of the team to tap directly into shore-based experts to assist in problem diagnosis, and to actually perform maintenance or error correction activities.

As mentioned above, Fleet personnel should be included in distributed networking exercises. Facilities such as the AEGIS Training Center, and the Fleet Combat Training Center, Atlantic and Pacific (FCTCLANT & PAC), when integrated on the net and with BFTT, will be able to provide more realistic training and have direct access to realistic scenarios and actual Fleet exercises.

Tactical situations that stress sensor and weapon capabilities can readily be accessed from the distributed data-base library. Fleet exercises that need to be scaled back due to cost considerations can be "force multiplied" and expanded to more closely represent real-world conditions.

Interoperability—Command, Control, and Communications

The expanding role of the Navy in Joint operations can be supported by participating in Joint network operations. These operations can aid in the assessment of Joint interoperability and battle management/command, control, communications, and intelligence (BM/C^3I) exercises. As briefly noted earlier, tomorrow's team of warriors will be even more dependent upon information than in the past. The inclusion of existing and notional tactical and strategic C^4I systems in the development process is an additional benefit of distributed networking.

Administrative and Acquisition Support

The ability to evaluate conceptual, competing systems on a "level playing field" *prior* to contract award will be a major result of a mature distributed networking capability. Program managers will be able to determine which proposed system(s) will best satisfy operational needs. Requests for Proposals will demand that bidding systems provide models and simulations along with responding quotes. The use of distributed technology is very attractive in this arena. The Federal Acquisition Streamlining act of 1994 calls for the creation of a government-wide Federal Acquisition Network (FACNET) by 1997. Prospective bidders will be able to "protect" their algorithms. Company secrets and innovative ideas will not be exposed; only integration protocol data will be passed via the net and interact on the evaluation "playing field." The same *types of networks* used by the acquisition community to develop and rapidly field new operational capabilities should also be used to increase the efficiency of the administrative bureaucracy. Review cycles can be shortened, reproduction and mailing costs reduced (if not eliminated), and the overall administrative infrastructure streamlined. The increased use of desktop video teleconferencing may also help reduce travel costs and the large number of meetings.

Engineering Development/Simulation-Based Design

The use of virtual models and simulations will support the development of new systems using tools such as enhanced computer-aided design/computer-aided engineering (CAD/CAE), (e.g., geometric shapes, weight, electric loads, center of gravity, flight kinematics, etc.). The software used to control robotics and numerical process control tooling may even have its genesis from this level.

Summary and Principles

Ensure a Joint Approach

Access to Joint systems and national resources (especially intelligence, reconnaissance, and other information-related assets) can be accomplished by distributed networking. Arrangements must be made to ensure that operational Navy assets are linked via both tactical and simulation communications networks. A standing Joint systems engineering test bed would be used to facilitate Joint engineering efforts. The aforementioned "bulletin board" of Joint models and simulations, combined with other Navy/DoD and Joint agencies/contractors (i.e., threats, environments, today's systems, etc.), should be used to help achieve this systems approach.

Model-Test-Model

Address complex systems engineering problems one step at a time. Use the network of today's resources as a starting point and incrementally build new or improved capabilities one element at a time. This will aid the critical *validation* task. Testing must address *technical* and *operational* performance of elements within the overall system of systems. The data obtained during "live" testing and exercises must be used to improve and verify/validate models, simulations, and processes.

Ensure Realism

Area and Theater scenarios that include the environment (i.e., terrain, weather, neutral players,

threats, etc.) must be factors. Again, the objective is to test the entire system, shipboard elements, joint interoperability, C^4I, tactics, etc. Maximizing the use of operator(s) in the loop at the appropriate times during the development process will ensure realism in the acquisition process.

Develop, Test, and Verify Systems Concurrently

The entire support structure of independent quality assurance and the two-stage T&E cycle (discrete Technical Evaluation (TechEval) and OPEVAL events) needs to be examined and *streamlined.* Quality assurance and testing must be continually conducted and incorporated at every stage of development. As noted above, operational suitability must also be *done early* in the process—not as the last step.

Minimize Throwaways

Evolve as much of the conceptual system into the tactical product as possible. Software developed to support/simulate early conceptual analyses must be considered for porting into the prototype and operational systems. The growing use of commercial items and open architecture designs supports this idea. Adapt software developed during the simulation process for use in the final product. Use this software in tactical systems as appropriate (crew training is an excellent first candidate).

Integrate Existing Resources

Leverage multiple sponsors and programs (Ballistics Missile Defense Operations (BMDO), Office of the Chief of Naval Operations (OPNAV), U.S. Air Force (USAF), U.S. Army (USA), U.S. Marine Corps (USMC), Overseas Supply Division (OSD), Advanced Research Projects Agency (ARPA), etc.). Integrate the products of these sponsors in a "common sense" fashion. Take advantage of *good*, existing development practices, eliminate inefficiencies, and streamline processes.

Flexibility

Programmatic milestones *must not be held hostage* to high-risk, developmental projects. These developing technologies will be addressed by allocating the appropriate resources to their exploration and solution. The overall plan must assure that the key technologies are in place at the appropriate time to support the mainline programmatic effort. The early inclusion of prospective technological advances may be addressed as noted in paragraph C, Part 1, of DoD Directive 5000.1, on aggressive prototyping and modeling and simulation.

Start Now

Identify a program or programs undergoing an upgrade and actually develop the conceptual prototype. Start by linking existing operational systems that this pilot program can interact with. Conduct an inventory of existing models and simulations. Use this catalog to identify which items are needed to support the program development (constructive and distributed). Create, via simulations, the desired new objects or portions of the desired system. Integrate these with the existing systems. Conduct evaluations of operational suitability and technical feasibility. The results of these evaluations may lead to further analysis, development, or program cancellation. The essential concept is to tackle only that which is affordable. The use of existing assets must be maximized.

Conclusion

System design and acquisition teams need the proper tools to address tomorrow's evolving requirements. Rapid prototyping, if properly integrated with a holistic Joint approach, can be a powerful tool for imaginative designers (Figure 10). The United States Army's network of multiple warfare area battlelabs is one example of such an integrated "toolkit." General Krulak, the Commandant of the Marine Corps, is another advocate. He has initialized the creation of the "Sea Dragon" battlelab at the Marine Corps Systems Command to address many of the issues discussed here. The Joint Chiefs of Staff are also pursuing similar developments, such as the Joint Battlelab for C^4ISR and the Joint Decision Support Center.

There is ". . . no systematic process in the Navy for thinking about major innovations in naval warfare or for speeding their introduction to the fleet."

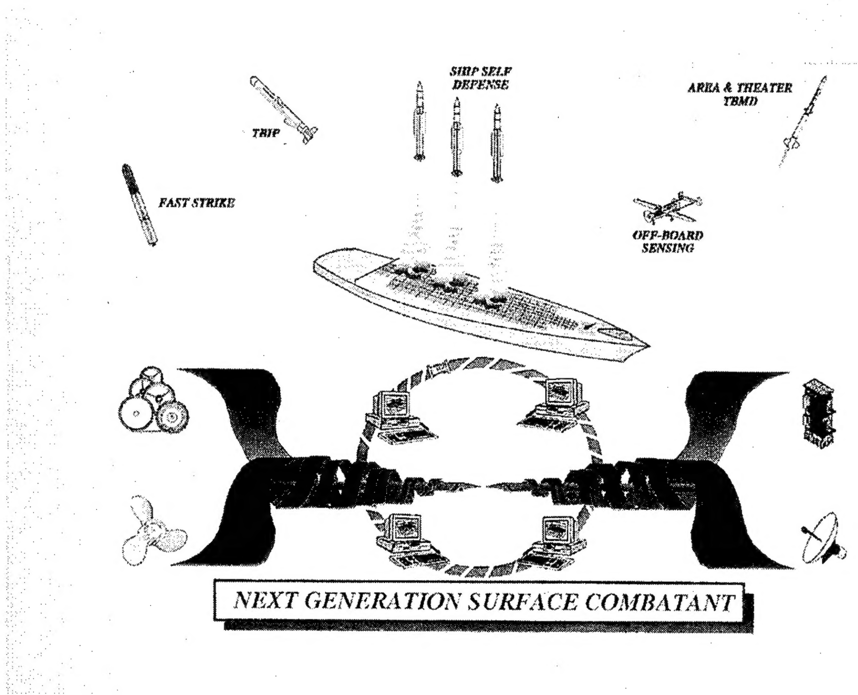
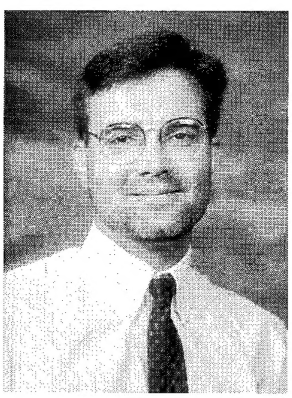


Figure 10. *Integrated system(s).*

The Author



ROBERT E. PODLESNY received a B.A. degree in criminal justice administration from the University of Maryland in 1977, a B.S. degree in computer science at the University of South Alabama in 1983, and an M.S. degree in systems technology from the University of Southern California in 1986. As a First Lieutenant in the U.S. Marine Corps from 1977 to 1981, he specialized in Tactical Air Control, Air Defense, and C^3 Systems. From 1983 to 1986, he was a computer scientist at Naval Coastal Systems Center working with Mine Warfare and Navigation Systems. While working as an electronics engineer at Naval Sea Systems Command from 1986 to 1989, he specialized in Antisubmarine Warfare and Surface Ship Torpedo Defense. From 1989 to 1991 he worked as a general engineer with the PEO for cruise missiles and unmanned air vehicles as well as serving as Branch Head of TOMAHAWK Systems Engineering. From 1991 to 1995, he has served as an electronics engineer at NSWCDD in various roles: TOMAHAWK Systems Engineering Branch Head from 1991 to 1993, Theater Air Defense Distributed Simulation Technical Coordinator from 1993 to 1994, and Director of Infrastructure Warfare Assurance from 1995 to the present.